
PhD Thesis

Statistical classification of images

Author:

María Andrea Giuliodori

Advisors:

Rosa Elvira Lillo

Daniel Peña Sánchez de Rivera



Department of Statistics

Universidad Carlos III de Madrid

Getafe, June 2011

d

*A Pascual y nuestra hija Milena, los dos grandes amores de mi vida y a
mis padres Saída y Roberto.*

Acknowledgements

In September 2004, I started the first year courses of the PhD program of Business Administration and Quantitative Methods¹. At that time, obtaining the doctoral degree was simply wishful thinking. It looked so far away. Seven year later, it has become a reality. Without question, it was one of most important projects of my professional life, the one I dedicated more time and effort. The pages of this dissertation capture only partially the knowledge acquired and the work done during these years. Yet, I hope I was able to summarize reliably the most relevant aspects.

The experience was more enjoyable and pleasant thanks to many people that accompanied directly or indirectly during this journey. First, I would to thank my advisers, Daniel Peña Sanchez de Rivera y Rosa Elvira Lillo Rodriguez, for the invaluable guidance, continuous support, and strong leadership. They have been extremely generous in sharing their knowledge with me and dedicating time and understanding when difficult times appeared. In many and different regards, they are role models to me. Thank you!

I would also like to acknowledge the support of the Department of Statistics at Universidad Carlos III de Madrid and all its members. In particular, I'm grateful to the Head of the Department Javier Prieto and Andrés Alonso, PhD director, for their kind concern and consideration regarding my academic requirements and for their continuous support during my motherhood. I also want to thank Mike Wiper, Belén Martín Barragan, Pedro Galeano, Concepción Ausín, Aurea Grané, Regina Kaiser, for helping me at different stages to understand the "secret" of statistics. Also, I'd like to acknowledge the help and support of the staff of the department, Francisco Saveedra and Susana Linares, really help make this place the success that it is.

I wouldn't have gone this far without my classmates: Anna, Ale, Santi, Adolfo, Alba, Júlia, Ana Laura, Maye, Ester, Paula, Peter y Pepa. With them I share the late nights, long meetings, and endless issues. They started simply as classmates but today

¹The authors gratefully acknowledge the support of "Comunidad de Madrid" grant CCG07-UC3M/HUM-3260 and of "Ministerio de Ciencia e Innovación" Acción integrada grant HI2008-0069

they are good friend. And I truly value their friendship.

I would also like to thank two institutions that hosted me during my visiting scholar periods: London School of Economic (UK) and Concordia University (Canada). My hosts Howell Tong (LSE) and Yogendra Chaubey (Concordia) made my stays an unforgettable experience.

I am also grateful to the members of the dissertation committee for their willingness to be present at my defense and provide insightful insights.

There may be other people that I'm forgetting in this writing but I offer my regards and blessings to all of those who supported me in any respect during the completion of this project.

A special thank goes to my family, both political and direct. My parents in law opened their doors and since day one, they made me feel as part of the family. Both Jorge and Graciela have been extremely helpful in supporting me with different actions in the consecution of this dissertation. I own them my deepest gratitude.

A special note also goes to my parents, Roberto and Saída. They have inculcated me the passion and love for the academic world. They unconditionally supported me when I decided to leave my country (and leave them) to embark in a PhD program. I know it was hard on them and yet they encouraged me to do it. They are an inspiration and a continuous source of admiration. My brother Alejandro and David, their wives and children have provided invaluable emotional support from the distance.

Last but not least, this thesis would not have been possible unless two persons were not in my life: my husband Pascual and our daughter Milena. They have helped in so many ways that I would probably need another dissertation to thank them. They gave me the strength to plod on despite my constitution wanting to give up and throw in the towel. Thank you, thank you so much. I love you both to death.

Abstract

Image classification is a burgeoning field of study. Despite the advances achieved in this camp, there is no general agreement about what is the most effective methods for the classification of digital images. This dissertation contributes to this line of research by developing different statistical methods aim to classifying digital images. In Chapter 1 we introduce basic concepts of image classification and review some results and methodologies proposed previously in the literature. In Chapter 2 we propose a method to classify images by their content. We are able to distinguish between landscape from non-landscape pictures by using three features obtained directly from images. We obtain better classification rates than those obtained by other authors dealing with similar kind of scene classification. In Chapter 3 we address the handwritten digit recognition. We suggest a set of intuitive features to perform the classification. Since the features are calculated with the binary image, we propose a novel technique to obtain the optimum threshold to binarize images, based on statistical concepts associated to the written trace of the digit. The classification is conducted by applying multivariate and probabilistic approaches, concluding that both methods provide similar results in terms of test-error rate (3.5%). In Chapter 4 we propose the application of Functional Data Analysis to analyze and classify images. While a limited number of authors have suggested the application of FDA for image classification [[Florindo et al. \(2010\)](#)], we suggest that this branch of statistics has represents a promising approach and offers several avenues for future research. We close the dissertation in Chapter 5 with a set of concluding remarks. Overall, the methods suggested in this dissertation are simple to apply, intuitive in their interpretation and their performance is comparable with other complex methods applied to the same problem. Moreover, the features suggested require less processing time than other methods (as support vector machine classifiers) and therefore require less computational capacity.

Resumen

La clasificación de imágenes es un campo de estudio de rápido crecimiento. A pesar de los avances logrados en esta área, no existe un acuerdo generalizado acerca de cuál es el método más eficaz para la clasificación de imágenes digitales. Esta tesis contribuye a esta línea de investigación mediante el desarrollo de diferentes métodos estadísticos que tienen como objetivo la clasificación de imágenes digitales. En el capítulo 1 se introduce los conceptos básicos de clasificación y se revisan algunos resultados de las metodologías propuestas previamente en la literatura. En el capítulo 2 se propone un método para clasificar las imágenes por su contenido. Somos capaz de distinguir entre una imagen de un paisaje de una que no lo es a partir del uso de tres variables obtenidas directamente de las imágenes. Obtenemos mejores tasas de clasificación que las alcanzadas por otros autores que han trabajado clasificación de escenas similares. En el capítulo 3 abordamos el reconocimiento de dígitos escritos a mano. Sugerimos una serie de variables intuitivas para llevar a cabo la clasificación. Dado que las variables se calculan con imágenes binarias, se propone una novedosa técnica para obtener el umbral óptimo para imágenes binarizadas, basado en los conceptos estadísticos asociados al trazo de escritura del dígito. La clasificación se lleva a cabo mediante la aplicación de métodos multivariantes y probabilísticos, concluyendo que ambos métodos proporcionan resultados similares en términos de tasa de error (3,5 %). En el capítulo 4 se propone la aplicación del Análisis Funcional de Datos para estudiar y clasificar imágenes digitales. Mientras que un número limitado de autores han sugerido la aplicación de ADF para la clasificación de la imagen [[Florindo et al. \(2010\)](#)], creemos que este rama de la estadística representa un enfoque prometedor y ofrece diversas alternativas para la investigación futura. Cerramos la tesis en el capítulo 5 con un conjunto de observaciones finales. En general, los métodos propuestos en esta tesis son fáciles de aplicar, intuitivos en su interpretación y su rendimiento es comparable con otros métodos complejos aplicados al mismo problema. Por otra parte, las características sugeridas requieren menos tiempo de procesamiento que otros métodos (como los clasificadores de técnicas de vector soporte).

Contents

List of Tables	XI
1. Introduction	1
1.1. Some basic concepts	4
1.2. Outline of the Thesis	6
List of Figures	1
2. Landscape image classification	9
2.1. Databases	12
2.2. Features	13
2.2.1. Local Variability	14
2.2.2. Effective Variance	15
2.2.3. Spatial Correlation	18
2.2.4. Feature selection	20
2.3. Supervised classification	24
2.4. Comparison to Support Vector Machine	31
2.5. Unsupervised classification	35

2.6. Conclusions and contributions	37
2.7. Future work	38
3. Handwritten Digit Classification	43
3.1. Databases	45
3.2. Binarization	47
3.3. Features extraction	49
3.3.1. Hough Transform	50
3.3.2. Euler number	58
3.3.3. Holes	60
3.3.4. Right and left entries	60
3.3.5. Cross in the center	60
3.3.6. Extremes	62
3.3.7. Intersections	62
3.3.8. Distance	63
3.4. Probabilistic classification approach	64
3.5. K nearest neighbor classification approach	66
3.6. Conclusions and contributions	68
3.7. Future works	68
4. Functional Data Analysis for images	71
4.1. Basis Functions	73
4.1.1. Fourier Basis	74
4.1.2. Haar Basis	75
4.1.3. Number of K basis	76
4.2. Smoothing and penalization	77
4.2.1. Smoothing by least squares	78
4.2.2. Smoothing with penalization	79
4.2.3. Smoothing with parametric penalization	79
4.3. Functional Principal Components	80
4.3.1. Application with Fourier basis	81

CONTENTS

IX

4.3.2. Application with Haar basis	85
4.3.3. Classification with functional principal components	87
4.4. Conclusions and future works	89
5. Final Conclusions and future works	91

List of Tables

2.1. ANOVA results	23
2.2. Classes distribution	24
2.3. Linear Discriminant Analysis results for RGB images	28
2.4. Linear Discriminant Analysis results for GRAY images	28
2.5. K Nearest Neighbor results for RGB images	30
2.6. K Nearest Neighbor results for GRAY images	30
2.7. K-MEANS algorithm information- <i>WLW</i> database	36
2.8. K-MEANS algorithm information- <i>GLP</i> database	36
3.1. Distributions of <i>MNIST</i> sets	45
3.2. Distributions of USPS sets	47
3.3. Missing straight lines	56
3.4. Probabilistic approach results	66
3.5. K- nearest neighbors results	67

List of Figures

1.1. RGB matrices	5
1.2. Different color-level of images	6
2.1. Indoor or outdoor scene?	10
2.2. Examples of images in <i>WLW</i> set	12
2.3. Examples of images in the <i>GLP</i> set	13
2.4. Maximum Local variability	15
2.5. Different values of Local variability	15
2.6. Different values of Effective variance	17
2.7. Extreme values of Effective variance	18
2.8. Image in different sizes	18
2.9. Different structures of Spatial correlation	19
2.10. Spatial correlation structure	20
2.11. Misclassified images	31
2.12. Support Vector Machine	33
2.13. Example of Periodogram for a texture image	40
2.14. Example of Periodogram for a texture image	40
2.15. Example of Periodogram for a scene image	41

3.1. Typical Images from <i>MNIST</i> sets	46
3.2. Typical Images from <i>USPS</i> sets	47
3.3. Examples of binarized digits	48
3.4. The same digit with different threshold values	50
3.5. Representation of a line	51
3.6. Normal representation of a line.	52
3.7. Hough Transform	53
3.8. Examples of lines	55
3.9. Cartesian plane	57
3.10. Hough transform	57
3.11. Examples of circles	58
3.12. Euler Number	59
3.13. Hole variable	60
3.14. Entry variables	61
3.15. Cross in the center variable	61
3.16. Examples of cross in the center variable	61
3.17. Examples of extremes variable	62
3.18. Examples of intersection variable	63
3.19. Example of distance variable	63
4.1. Fourier basis	74
4.2. Haar wavelet	76
4.3. Example 1	82
4.4. Example I: four <i>FPCA</i> -Fourier basis	83
4.5. Example I: <i>FPCA</i> with penalization-Fourier basis	84
4.6. Example 2	84
4.7. Example II: <i>FPCA</i> with penalization-Fourier basis	85
4.8. Example 3	86
4.9. Example III: <i>FPCA</i> , comparison between Haar and Fourier basis	86
4.10. First functional principal component: measures	88
4.11. First functional principal component: modified measures	89

CHAPTER 1

Introduction

In our daily lives, we are able to recognize and distinguish an innumerable amount of information by simply observing our environment. For instance, we know whether it is winter, spring, summer or fall by only observing the leaves of trees, or we can guess the origin of a letter by looking at its zipcode. For years now, researchers have tried to equip devices with the same ability by developing different algorithms that process the quantitative information of images. From the simple internet search of images to the diagnosis of lethal diseases, digital image analysis is gaining importance as a valuable tool to recognize, order, and classify visual content. Despite the achievements obtained thus far, this kind of tasks is still a challenge since there is no agreement about which methods are the most appropriate. This thesis attempts to shed light on this issue as it deals with the statistical classification of images. We propose different multivariate methods to classify images using statistical measures to describe their content, such as to capture the variability in colors or the shape of objects they contain.

The treatment of digital images is not a new phenomenon. One of the first applications of digital images techniques was in the 1920s by the Bartlane cable picture transmission system. This system was named after Harry G. **Bar**tholomew and May-

nard D. McFarlane and was developed in Great Britain. The technique allowed that digitized newspaper pictures that were sent by submarine cable between London and New York. Pictures were coded for cable transmission and then reconstructed at the receiving end on a telegraph printer. The early Bartlane systems were capable of coding images in five distinct gray levels. This was increased to fifteen levels in 1929. Introduction of the Bartlane cable picture transmission system in the early 1920's reduced the time required to transport a picture across the Atlantic from a week to less than three hours. Although it was an advance at the time, the resulting pictures looked like small, embroidered black and white pieces of paper. It is not until the 1950s and 1960s when improvements in computer technology led to a surge of work in digital image processing. During the 1970s the so-called mathematical morphology theory [Serra (1984)] arose for the analysis of geometrical structures based on set theory, lattice theory, topology, and random functions. This methodology is most commonly applied to digital images analysis in geology and biology fields [see Castleman (1995), Duda and Har (1973), Duda et al. (2000), Pratt (1991) and Serra (1984)]. Later, digital image processing begins to be used in medical applications by the invention of tomography. The mayor contribution in the 1980s is the development of algorithms to detect characteristics like edge, lines [Hough transform, Duda and Hart (1972)] and textures in images. Besides, the introduction of environment information to reconstruct scenes acquired importance in the classification and segmentation of images [see Besag (1986), Cross and Jain (1983) and German and Geman (1984)]. With the fast computers and signal processors available in the 2000s, the analysis of digital images has become the most common form of image processing and, generally, is used because it is not only the most versatile method, but also the cheapest.

Today, the applications of digital image analysis are continuously expanding through all areas of science and industry, and it is used to solve different kind of problems. For example, in medicine it is used to detect diseases as cancer, through a magnetic resonance imaging scan. In biometry, it is used to the recognition of faces, iris, handwritten and fingerprints, sometimes for security reasons. The increasing number of applications that may use digital images creates the necessity to classify them by an easy and

fast way. Besides, there are applications that can benefit from image classification. For instance, the commonly used internet image search engine has poorly performance in finding images due to the use of filenames instead of some content-based classification method ¹. Therefore, describing images by their content (rather than names) constitutes an useful way to organize and group them to facilitate searches.

Image classification methods can be roughly divided into two approaches. On the one hand, there are learning-based classifiers that require an intensive learning phase of the classifier parameters. Nowadays, these techniques have attracted growing attention for their wide applicability. Examples of these classifiers are neural networks [[Shah and Gandhi \(2004\)](#), [Ciresan et al. \(2010\)](#)], radial basis functions [[Yuchun \(1991\)](#)], support vector machine [[Varma \(2007\)](#), [Zhang et al. \(2007\)](#)], boosting [[Liu et al. \(2004\)](#)], learning vector quantization [[Thulasiraman \(2005\)](#)] and decision trees [specially random forests, [Bosch et al. \(2007\)](#)]. On the other hand, there are classifiers that do not require the learning process, such as the kernel estimation classifiers (specially k-nearest neighbor) and the linear discriminant classifiers. These methods also report good performance in classification.

Although there is a large number of classifiers available, there is no agreement regarding which method is most appropriate. The reason is that a method works better or worse depending on the database used. Despite the contributions and advances produced thus far, the field still struggles to find fast and simple methods to perform the classification easily. This dissertation attempts to contribute to the classification of digital images by proposing different statistical methods that, compared to other approaches, are easier to calculate, more intuitive and more generalizable to other databases.

¹Google has a recent Beta-version application to search images with similar content. However, this search is still restricted to a limited set of images.

1.1. Some basic concepts

In statistical classification of images a picture is associated with a group of variables or measures extracted from it and used to perform the analysis. The measures used for classification in this thesis are called indistinctly *features* or *variables*. In the general case, t features x_i , for $i = \dots, t$, are used, forming the *feature vector* given by

$$x = (x_1, x_2, \dots, x_t).$$

Each feature vector identifies a single image. In this work, features and feature vectors are treated as *random variables* and *random vectors*, respectively. Then, the goal in classification is to find the feature vector which best differentiate two or more categories. The effectiveness of the feature vector is determined by how well images from different classes can be separated, and it is measured by the error rate. There are different definitions of error rate commonly used in the literature. In this thesis we define the error rate as a percentage of misclassified images².

A digital image can be observed in different levels: gray-level, color-level, and binary-level. The term gray level image is related with a two-dimensional function $f(x, y)$, where x and y denote spatial coordinates and the value of the function f at any point (x, y) represents the gray level of the image at that point. The x axis is usually the horizontal axis, while the y axis is usually the vertical axis of the image. Then, the function $f(x, y)$ is discretized to be represented as a matrix, whose row and column indices identify a point in the image. The corresponding matrix element values identify the gray level at each point and are called image elements, picture elements or simply pixels.

An image in color-level is represented in different models of color. We use one of the most applied and developed by computer programs, that is, the RGB model. The system RGB is based in human perception of colors and its name comes from the primary colors produced by light, **R**ed, **G**reen and **B**lue. This mixture is called additive

²In learning-based works the term error rate is used as synonyms of rejection error rate. That is, an image is rejected to be classified in a class whenever the classification cannot be achieved with enough confidence [Bottou and Vapnik (1992)].

synthesis, and any combination of intensity of the three colors produces the majority of colors in the visible spectrum. The main purpose of this model is the representation and display of images in electronic systems such as computers and televisions. The representation of a color-level digital image is given by a 3-dimensional matrix (R,G,B), constructed through the three primary colors. The tone of a particular pixel of a picture is obtained by the combination of the information (red, green and blue values) contained in each dimension (see Figure 1.1). Typically, pixel values vary between (0,255). However, we change this scale to the range (0,1) for simplicity. The color of a specific pixel in the position (x,y) will be $(R(x,y), G(x,y), B(x,y))$. For instance, a pixel with values $(R(x,y), G(x,y), B(x,y)) = (0,0,0)$ corresponds with a white pixel. In contrast, a pixel with values $(1, 1, 1)$ corresponds with a black pixel.

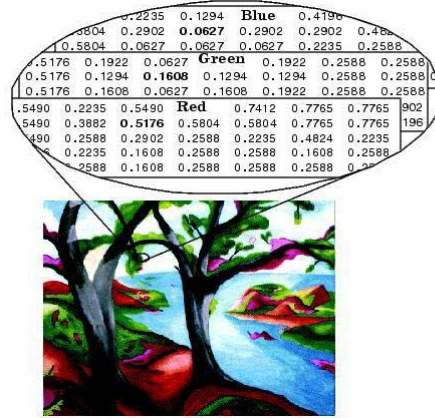


Figure 1.1: RGB matrices

An image can be transformed from RGB level to gray-level, that is, from a 3-dimensional matrix in RGB colors to a 2-dimensional matrix in gray level. This conversion is established by the international norm for digital TV (CCIR-601). Each gray pixel is obtained from RGB pixels through the following linear combination

$$graypixel(x,y) = 0.299 \times R(x,y) + 0.587 \times G(x,y) + 0.114 \times B(x,y), \quad (1.1.1)$$

where $R(x,y)$, $G(x,y)$ and $B(x,y)$ are the value of the pixel (x,y) in the red, green and blue matrices respectively. The gray conversion arose during the 1930s as a necessity

to combine the white and black television with the model of color used. Then, the concept of *lightness* emerges to carry the information of bright and light of images, that is the white and black color (or lightness component).

Finally, a binary image is a digital image that has only two possible values for each pixel. Generally, the two colors used for a binary image are black and white, but any two colors can be used. The color used for the object(s) in the image is the foreground color, while the rest of the image is the background color. A binary white and black image is the result of a thresholding operation applied to the gray level image. That is, given a threshold the binary image has pixels with one value if they have the gray intensity greater than the threshold, and pixels with value equal to zero if they are lower than the established threshold in gray level image. Examples of the three levels are given in Figure 1.2.



Figure 1.2: Different color-level of images

1.2. Outline of the Thesis

This thesis addresses the classification of images by statistical techniques, that is, given a set of images, our goal is to classify them into categories. This task is subject of many recent works [Ayers and Boutell (2007), Bosch et al. (2007), Ciresan et al. (2010), Gómez (2009), Liu et al. (2007), Nandgaonkar et al. (2010), Patino-Escarcina

and Ferreira Costa (2008) and Qin and Yung (2010)], and tackled by different approaches. We will use statistical techniques, based on a combination of features which show a good performance in classifying images.

There are at least two main branches that have images as objects of analysis: computer vision and pattern recognition. There is a significant overlap in the range of techniques and applications in both approaches. For instance, they share the problem to determine whether or not the image data contains some specific object, feature or activity. However, in computer vision the principal goal is to build a system that can understand images as well as a human, while pattern recognition aims to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. Chapter 2 and 4 of this thesis are closely related with computer vision applications, while Chapter 3 is more connected with applications in pattern recognition.

The thesis is structured as follows. Chapter 2 is devoted to the classification of scene pictures. We describe four features to represent the variability and dependency structure of pixels in the image, that show discriminative power to classify images of landscape and non-landscape scenes. The analysis is done over two databases in color-level. Additionally, we obtain results for the images converted to gray-level. The classification is conducted by supervised methods, i.e., Linear Discriminant and K-Nearest Neighbor classifiers. We obtain results comparable with complex methods applied for the same issue. In Chapter 3 we study the classification of scanned handwritten digit using three different databases. Although the datasets contain grayscale digits, they were transformed to binary level since the features used require binary images. We carry out the classification comparing the results of the K-nearest neighbors algorithm with a probabilistic approach based on the application of the Bayes's rule. We obtain similar conclusions in both cases. Moreover, in this chapter we propose a novel method to find the optimum threshold to binarize the digit. In Chapter 4, we initiate the uses of Functional Data Analysis to classify images. We perform a preliminary study over the scene databases (landscape & non-landscape). Finally, we close this dissertation with a set of conclusion included in Chapter 5.

CHAPTER 2

Landscape image classification

Computer vision is a discipline that extract information from images by automatic techniques. An emerging research field in computer vision is content-based image organization and retrieval (CBIR). One of its challenges is the classification of image scenes (or simply scenes) using methods that best reproduce the human concepts and thought. Several obstacles limit the success of this classification: the wide variety of scenes that describes each category; the illumination, the camera uses to obtain the pictures and other factors that introduce noise into scenes. Besides, the human visual classification of scenes has certain subjectivity depending on whoever makes the observation of that scene. For example, Figure 2.1 shows a picture that can be classified as *indoor* or *outdoor* scene depending on the individual interpretation of the meaning *outdoor* and *indoor*. For that reason, there is an important semantic gap between the classification by automatic techniques and the classification carried out by the subjective human thought. Consequently, research in this area is focused on diminishing this gap. In order to tackle this problem, there are two strategies commonly used in the literature to perform the scene classification. The first one uses low-level features extraction. The low-level features are referred to the characteristics extracted

directly from the image, considering the scene as an individual object. Examples of these features are the texture, color, geometric parameters, and statistics of color and tones. Frequently, they are used in contexts with small number of categories. Previous work on this approach [e.g., [Szummer and Picard \(1998\)](#)] achieved a classification rate around 90% in indoor-outdoor classification by using the K-nearest neighbor classifier. [Vailaya et al. \(1998\)](#) and [Vailaya et al. \(2001\)](#) tried to capture some concepts by the low-level features extraction, reporting classification rates of 90.5% in indoor-outdoor classification, 95.3% in city-landscape, 96.6% in sunset-forest & mountain, and 96% in forest-mountain classification. Other references of this approach are [Ayers and Boutell \(2007\)](#), [Huang et al. \(2007\)](#) and [Nandgaonkar et al. \(2010\)](#).



Figure 2.1: Indoor or outdoor scene?

The second strategy is related with the use of high-level features that requires an intermediate analysis of the images before the classification. The high-level concepts are used by learning-based techniques trying to reproduce the human thought in classification process. They are often applied to situations with large number of scene categories. An example of this approach is the reference of [Bosch et al. \(2007\)](#), who detected objects into the scenes which helped in the classification. They used the unsupervised probabilistic Latent Semantic Analysis followed by the K-nearest neighbor classifier to perform the classification. [Qin and Yung \(2010\)](#) used contextual information (related words) to reduce the ambiguity of some pictures. Authors used three databases with 8, 13 and 15 categories, obtaining classification rates of 90.30%, 87.63% and 85.16% respectively. [Park et al. \(2004\)](#) used neural network techniques

on two different databases with 30 classes each, achieving classification rates of 81.7% and 76.7%. Other references of this approach are [Belongie et al. \(1997\)](#), [Boutell et al. \(2004\)](#), [Liu et al. \(2007\)](#), [Shah and Gandhi \(2004\)](#) and [Wang et al. \(2001\)](#).

Although the two strategies are applied to different contexts, some authors integrated both in an attempt to reduce the semantic space between the content of images and the features extracted from them. For instance, [Luo and Savakis \(2001\)](#) used a Bayesian network to combine the knowledge of low-level and high-level features, obtaining a classification rate of 90.1% for indoor-outdoor classification. [Serrano et al. \(2002\)](#) used Support Vector Machine to classify indoor-outdoor scenes obtaining a classification rate of 90.2%.

This chapter deals with the classification between landscape and non-landscape images scenes. Due to the low number of classes, we follow the strategy based on low-level features extraction to classify these scenes. To this end, we consider three statistical features that summarize useful information of the image. We perform the classification on two different databases applying supervised classification (K-nearest neighbor algorithm and the Linear Discriminant classifiers) to split landscape from non-landscape scenes. The supervised classification is based on a priori knowledge of the membership class of a group, used to classify another group with unknown membership classes. We obtain classification rates around 97% and 95%. These error-rates improve the best methods reported in the literature for this kind of problem. Finally, we conduct an unsupervised classification to form groups, in order to confirm the existence of two different sets in the databases. In unsupervised classification there is not prior knowledge about the membership class of any element in the set. Example of this is the k-means algorithm. Procedures are executed using color and grey levels and results are later compared. We also perform the classification applying other competitive procedures such as support vector machine classifier obtaining better performance in our case.

This chapter is organized as follows. First, we describe the databases used in the classification in Section 2.1. In Section 2.2, we describe the three statistical low-level features. The first one is a measure of local variability obtained by the spatial changes

of the pixels in the image. The second one is a measure of global variability based on the singular value decomposition of the image matrices. The third variable measures the spatial correlation among the pixel intensities. In each section we analyze the relevance of the features and characteristics obtained in order to perform the classification of landscape vs- non-landscape image scenes. Section 2.3 and 2.5 are devoted to classify the images by the application of supervised and unsupervised techniques. Comparisons of results with other techniques are provided in Section 2.4. Finally, Section 2.6 gives some conclusion and discusses different avenues for future work.

2.1. Databases

In this experiment we use two databases. One of them is a subset of 379 images in color level, extracted from the collection of Wang, Li and Wiederhold pictures [Wang et al. (2001)]¹. We call it *WLW* set. The criteria to select the subset was to include images with the same size (128×96) in order to avoid possible distortions in the proposed measures caused by the resizing methods. The set contains 213 landscape images, and 167 pictures of other scenes as foods, buildings and monuments. Figure 2.2 includes some samples of this dataset.



Figure 2.2: Examples of images in *WLW* set

The other database is a group of images specially collected for this dissertation

¹<http://wang.ist.psu.edu/docs/related/>

from internet, by using the search engine *Google Images*. We call this database *GLP* (Giuliodori-Lillo-Peña). It consists of 379 heterogeneous color-level images randomly collected, with similar size than those in *WLW*. This set includes 85 landscape pictures, and 294 images of other scenes, like faces, ID cards, babies, animals and paintings (see Figure 2.3).



Figure 2.3: Examples of images in the *GLP* set

After making a visual inspection of both databases we observe that the *GLP* database seems to be more heterogeneous than *WLW* set, that is, contents of images in *GLP* appear to be more varied. For instance, there are people, animals and textures in this dataset, which cannot be found in the *WLW* set. Some of the following results will be influenced by this heterogeneity.

2.2. Features

In this chapter we propose the use of three low-level features, that is, three characteristics extracted directly from the image to conduct the classification. These variables describe the variability and spatial correlation of images. They are: the local variability, the effective variance, the spatial correlation and the spectral density function. The analysis of the spectral density function, is later dismissed in the classification process because do not show discriminant power among the classes we aim to classify.

2.2.1. Local Variability

The first variable proposed is the local variability, denoted as $\delta(X)$. This feature, introduced by Benito (2006), is a smoothing measure that represents the spatial dependence within pixels. Given an image X of size $n \times m$, the local variability is obtained through the bi-dimensional derivative and is defined as follows. Given a pixel x_{ij} , the estimation of the derivative of the intensity value in this point, is defined as

$$\nabla_{ij}(X) = x_{i+1,j+1} - x_{i+1,j} - x_{i,j+1} + x_{i,j}.$$

Then, the local variability is the result of

$$\delta(X) = \frac{1}{\tilde{d}} \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} |\nabla_{ij}(X)| ; \text{ for } i = 1, \dots, n \text{ and } j = 1, \dots, m \quad (2.2.1)$$

where $\tilde{d} = (n-1) \times (m-1)$ and the notation $|\bullet|$ represents the absolute value. Higher values of δ are observed in images with abrupt changes of intensity. The maximum local variability is achieved when the derivatives are maximum. For standardized values of pixel x_{ij} (in the range $[0,1]$), the derivative of a pixel is maximum when the neighbor pixels are completely different. For example, given the elements of an 2×2 matrix as follows,

$x_{i,j}$	$x_{i,j+1}$
$x_{i+1,j}$	$x_{i+1,j+1}$

considering abrupt changes in the neighbors, the previous matrix would look like

1	0
0	1

Henceforth, if derivatives are maximum, the value of the local variability is $\delta = 2$. In Figure 2.4 we generate an example of picture with maximum local variability.

In the datasets, we observed that images with high variability in colors show greater local variability. As it is explained in next section, this measure is calculated for the three RGB matrix in a color images. In Figure 2.5a it is shown an image with $\delta = (0,0511, 0,0511, 0,0511)$ for the RGB matrices, while in Figure 2.5b the values for the image are $\delta = (0,0154, 0,0154, 0,0154)$.

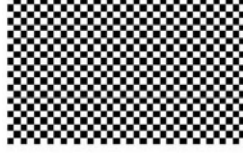


Figure 2.4: Maximum Local variability

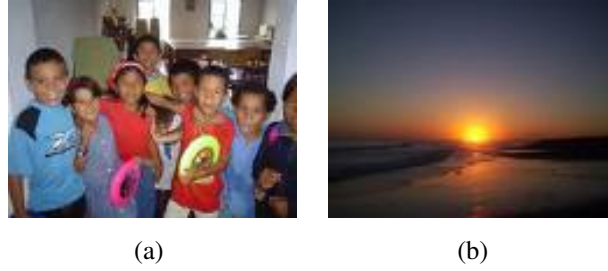


Figure 2.5: Different values of Local variability

2.2.2. Effective Variance

The second variable that we propose is the effective variance. This concept is introduced by [Peña and Rodriguez \(2003\)](#) to compare the variability in set of variables with different dimensions. Specifically, let X be a matrix with dimension $n \times m$ where its rows are the observations and its columns are the variables. The covariance matrix of X , denoted as Σ_X , is given by

$$\Sigma_X = n^{-1}(X - \mathbf{1}\bar{x}^T)^T(X - \mathbf{1}\bar{x}^T); \quad \text{for } m < n,,$$

where \bar{x} is the vector of means by columns of X , $\mathbf{1} \in \mathbb{R}^n$ and it is a vector of ones. Then, the effective variance is obtain as

$$EV(X) = (\phi_1 \phi_2 \dots \phi_p)^{1/p} \tag{2.2.2}$$

where $p = \min(n, m)$ and $\phi_1 \geq \phi_2 \geq \dots \phi_p$ are the eigenvalues of the matrix Σ_X . Subsequently, this measure is adapted by [Benito \(2006\)](#), to consider the high dimensionality of data in image analysis. The author introduced the concept of effective range to avoid a large proportion of zero or small eigenvalues, frequently found in images due to their

structure. The effective range, r_e , represents the number of eigenvalues (k) for which a relative error of the matrix reconstruction (left term in the inequality of equation 2.2.3) is less than ε . It is obtained as

$$r_e(X) = k = \{\#\phi / \frac{\|X - \hat{X}_k\|}{\|X\|} < \varepsilon\}, \quad (2.2.3)$$

where $\|\bullet\|$ represents the norm, the value of ε is a sufficiently small value, and \hat{X}_k is obtained by the singular decomposition of X using only the k greatest eigenvalues (ϕ) and their respective eigenvectors. That is,

$$\hat{X}_k = V_k D_k^{1/2} U_k^T,$$

where V_k and U_k are orthogonal matrices and their columns are the k eigenvectors of matrices XX^T and $X^T X$, respectively and D_k is a diagonal matrix whose elements are the k th greatest eigenvalues of XX^T or $X^T X$ (are the same). Using the L^2 norm, we write

$$\|X - \hat{X}_k\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^m (x_{ij} - \hat{x}_{ij})^2}.$$

The procedure in equation (2.2.3) is repeated until we find the value of k for which the relative error is less than ε . This final k represents the effective range r_e . The parameters in equation (2.2.3) are selected in order to include the greater non-null eigenvalues. We have used the following norms to obtain the effective variance in order to compare classification results with each one. Let A be a matrix with elements a_{ij} ,

- Infinity norm: the largest row sum of a matrix,

$$\|A\|_{\infty} = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|.$$

- 1-norm: the largest column sum of the matrix.

$$\|A\|_1 = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|.$$

- Frobenius-norm (also known as Hilbert-Schmidt),

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2}.$$

- Euclidean norm: the square root of the largest eigenvalue of the positive-semidefinite matrix $A^T A$,

$$\|A\|_E = \sqrt{\lambda_{\max}(A^T A)}.$$

We deduce that in the particular case of this norm, the effective range r_e in equation (2.2.3) is reduced to

$$r_e = k = \{\#\phi / \frac{\sqrt{\phi_{k+1}}}{\sqrt{\phi_{\max}}} < \varepsilon\},$$

Finally, the effective variance in (2.2.2) is expressed as

$$EV(X) = (\phi_1 \phi_2 \dots \phi_{r_e})^{1/r_e} \quad (2.2.4)$$

where ϕ_i are the eigenvalues of the matrix X .

In the databases we observe that pictures with variability in colors show high effective variance. In Figure 2.6 there are two examples with different values of EV . In Figure 2.6a the values for the three matrices are $EV = (0.01, 0.01, 0.008)$, while in Figure 2.6b the values are $EV = (0.04, 0.02, 0.01)$.



Figure 2.6: Different values of Effective variance

Besides, this measure assumes extreme values, for example in a completely random image with standardized pixels (in the range $[0, 1]$). In this case, the effective variance is equal to 1 (see Figure 2.7a). Conversely, for an image with pixels of the same color, the effective variance assumes the value 0 (see Figure 2.7b).

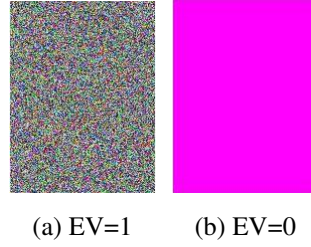


Figure 2.7: Extreme values of Effective variance

A weak point of the effective variance is that it changes when the image is resized. In Figure 2.8a we can see an image in its original size. The effective variance is equal to $EV = (0.69, 0.71, 0.67)$ for the RGB matrices. In Figure 2.8b there is the same picture scaled to 1.5 times the size of the original one. Now the effective variance is equal to $EV = (1.30, 1.31, 1.22)$. Thus, changes the size of the pictures leads to changes in the values of effective variance.



Figure 2.8: Image in different sizes

2.2.3. Spatial Correlation

The major part of the visual contribution of a single pixel to an image is redundant and can be inferred from its neighbors. This implies that there exists relative dependence between a pixel and its neighbors. In this context arises the concept of spatial correlation [Ripley (2004)], that is the third variable proposed in this work. This measure represents the correlation between one pixel and its neighborhood located to a distance h , where h assumes values from 1 to 15 (called spatial correlation of order h).

The spatial correlation is denoted as $\rho_h(X)$ and the values of h are defined in accordance with the size of the images. The calculation is done for each pixel, considering the ordination per row. Given a matrix X with size $n \times m$, the simplest version of the spatial correlation is defined as:

$$\rho_h(X) = \frac{\sum_{i=1}^n \sum_{j=1}^m [\sum_{k=-h}^h \sum_{t=-h}^h (x_{ij} - \bar{x})(x_{i+k,j+t} - \bar{x})]}{\sum_{i=1}^n \sum_{j=1}^m [\sum_{k=-h}^h \sum_{t=-h}^h (x_{i+k,j+t} - \bar{x})^2]}, \quad (2.2.5)$$

where x_{ij} is the element ij of the matrix X with range of variation $[0,1]$, \bar{x} is the mean of all pixels in the X matrix, $x_{i+h,j+h}$ are the values of the neighbor located to a distance h from x_{ij} , and h is the order of the spatial correlation, i.e., the distance between x_{ij} and its neighbors.

After calculating this measure, we detect some characteristics in the databases. In general, the spatial correlation of the images decreases as the order h increases. Another characteristic is that the three RGB matrices often show similar spatial correlation structure. Moreover, typical behaviors of spatial correlation are observed. For instance,

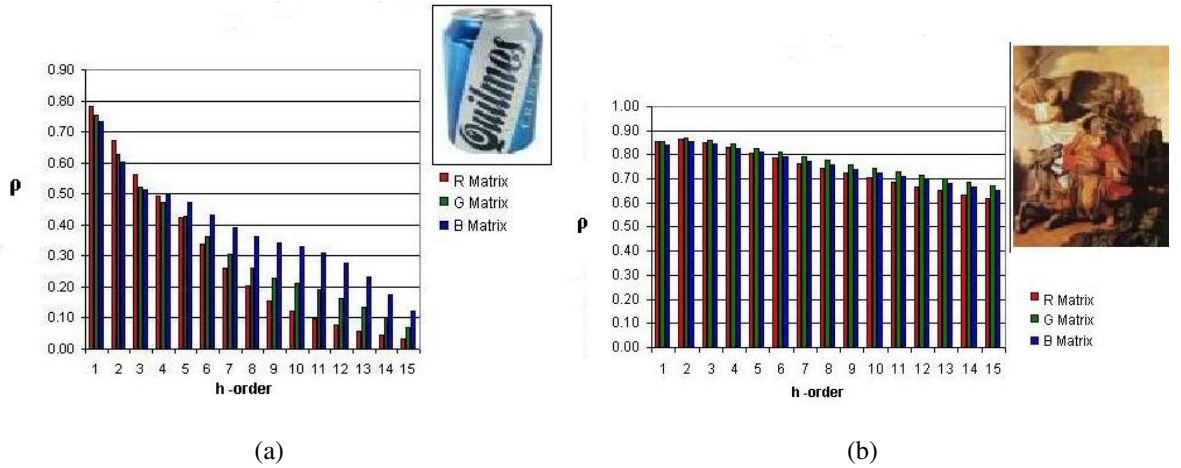


Figure 2.9: Different structures of Spatial correlation

some images have high correlation at the first order and it decreases slowly along the h-order (Figure 2.9b). Other images show a smaller spatial correlation at the first order ($h = 1$) and even smaller value at greater order of h (Figure 2.9a). Lastly, we observe an intermediate situation between the previous ones (see Figure 2.10). In the latter,

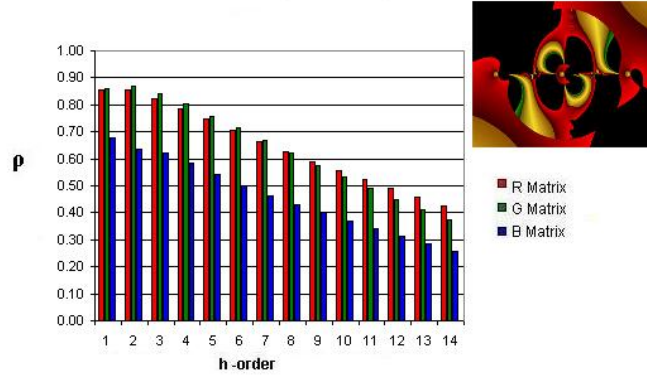


Figure 2.10: Spatial correlation structure

there are also small differences among the correlations of the three RGB matrices. It seems that the spatial correlations of middle orders are different in the three frequently detected structures. As a conclusion, one interpretation of these results may be that middle orders of spatial correlation could have discriminant power among images with different structures of this measure.

2.2.4. Feature selection

In the classification we use color-level and gray-level images in order to compare both results. Remark that our goal in this chapter is to use the features described previously to classify images as landscape or non-landscape. Next, we describe the variables considered to perform the classification.

In previous section we described four features to analyze the behavior in different group of images. However, considering the classes we aim to classify, the spectral density function do not show discriminant power between them. Thus, the variables used in the classification are the local variability, the effective variance and the spatial correlation.

For color-level images we calculate the variables for the three RGB matrices independently. Firstly, we calculate the local variability for every picture in each database (379 images each one). We denoted as δ_{Rr} , δ_{Gr} and δ_{Br} the local variabilities of matri-

ces R, G and B respectively, for an image r , where $r = 1, \dots, 379$. Therefore, we have a three-sized vector for each image.

$$\delta_r = (\delta_{Rr}, \delta_{Gr}, \delta_{Br}).$$

Afterwards, we calculate the effective variance of the three RGB matrices for every image of the datasets. The value of ε and k were chosen to include the number of eigenvalues that explained at least the 90% of the variability of the matrix, that is an $\varepsilon = 0.07$ and an initial value of $k = 0.3 \times \min(n, m)$. We also selected the Frobenius-norm because showed better performance in the classification.

The result for every picture is a three-sized vector given by

$$EV_r = (EV_{Rr}, EV_{Gr}, EV_{Br}).$$

where EV_{Rr} , EV_{Gr} , and EV_{Br} are the effective variances of the matrix R, G and B respectively, for the image r .

The third variable, the Spatial Correlation, is obtained for the RGB matrices and for values of $h = 1, \dots, 15$. We get a vector of 15 elements for each RGB matrix of every image, i.e.,

$$\rho_{Rr} = (\rho_{R1r}, \rho_{R2r}, \dots, \rho_{R15r}),$$

$$\rho_{Gr} = (\rho_{G1r}, \rho_{G2r}, \dots, \rho_{G15r}),$$

$$\rho_{Br} = (\rho_{B1r}, \rho_{B2r}, \dots, \rho_{B15r}),$$

where ρ_{Rhr} , ρ_{Ghr} and ρ_{Bhr} are the spatial correlation of order h for the matrix R, G and B respectively, of the image r .

Finally, for every image we obtained a feature vector of size 51 composed by the effective variances, local variabilities and spatial correlations (for $h = 1 \dots, 15$) for the three RGB matrices. Through the analysis of data we observed that some variables presented similarities in values in the three RGB matrices of the image. For instance, the local variability seemed to be the same for the three RGB matrices of all images. Therefore, in order to confirm this intuition, we performed an analysis of variance by considering the images of the dataset as the observations and the values of local

variability in the R, G and B matrices as the three groups. The hypothesis for local variabilities comparison were the following,

$$\begin{aligned} H_0 : \bar{\delta}_R &= \bar{\delta}_G = \bar{\delta}_B \\ H_1 : \exists \text{ some } \bar{\delta} &\text{ different,} \end{aligned}$$

where $\bar{\delta}_R$, $\bar{\delta}_G$, and $\bar{\delta}_B$ are the average local variabilities (for all the pictures in the database) for the matrix R, G and B respectively. We assumed that the distributions of the residuals are normal and the cases are independent. As a result, we obtained a *p-value* equals to 0.951 for *WLW* database and 0.964 for *GLP* set (see Table 2.1). That is, there was no evidence of meaningful difference among the means of local variability for the three RGB matrices. Then, we calculated the mean of local variabilities to use it as discriminative variable in the classification. It is obtained as follows,

$$\bar{\delta}_r = \frac{\delta_R + \delta_G + \delta_B}{3}. \quad (2.2.6)$$

Besides, we also observed that the spatial correlations presented similarities among the RGB matrices in some orders. Then, the analysis of variance was replicated for the rest of variables in order to reduce the information.

In the case of the effective variance, the analysis suggested that there was meaningful difference among the means of the effective variance for the three matrices with a *p-value* equals to 0.002 in both databases. Thus, we kept the effective variances of the three matrices for the classification. With respect to the spatial correlation, the analysis of variance showed that there was no evidence of significant difference among the mean correlations of the three RGB matrices for orders $h = 8, 9, 10, 11, 12, 13, 14$ and 15 in both databases. With respect to the spatial correlations of order $h = 1, \dots, 7$, they did not show the same behavior in both sets. Whereas in *WLW* dataset, there was evidence of significant difference among the mean spatial correlations of the RGB matrices for these orders, in *GLP* dataset did not. Consequently, we did not reduce the information of the spatial correlations of orders $h = 1, \dots, 7$, due to the different outcomes obtained in the databases. Therefore, we kept for the classification the following

Table 2.1: ANOVA results

Variable	WLW database	GLP database
	Significance p-value	
Effective Variance	0.002	0.002
Local Variability	0.951	0.964
Spatial Correlation 1	0.000	0.401
Spatial Correlation 2	0.000	0.283
Spatial Correlation 3	0.000	0.361
Spatial Correlation 4	0.002	0.508
Spatial Correlation 5	0.010	0.660
Spatial Correlation 6	0.030	0.783
Spatial Correlation 7	0.064	0.856
Spatial Correlation 8	0.109	0.901
Spatial Correlation 9	0.160	0.934
Spatial Correlation 10	0.216	0.961
Spatial Correlation 11	0.273	0.982
Spatial Correlation 12	0.333	0.993
Spatial Correlation 13	0.391	0.993
Spatial Correlation 14	0.455	0.994
Spatial Correlation 15	0.532	0.996

information about the spatial correlations.

$$\begin{aligned} \rho_{hr} &= (\rho_{Rhr}, \rho_{Ghr}, \rho_{Bhr}), \quad \text{for } h = 1, \dots, 7. \\ \bar{\rho}_{hr}, & \quad \text{for } h = 8, \dots, 15, \end{aligned} \quad (2.2.7)$$

where

$$\bar{\rho}_{hr} = \frac{\rho_{Rhr} + \rho_{Ghr} + \rho_{Bhr}}{3} \quad (2.2.8)$$

After reducing the number of variables applying the ANOVA test, we keep for each color-level image, 33 features out of the initial 51.

In order to study the performance of the classification in gray-level images, we made the conversion from color-level (RGB) to gray-level (see equation 1.1.1 of Introduction). Afterwards, we calculated the variables for the gray-level matrix of each

image. As result, we obtained the following information: the effective variance EV , the local variability δ , and the spatial correlation for orders $h = 1, \dots, 15$ given by $\rho_r = (\rho_{1r} \rho_{2r} \dots \rho_{15r})$. Those variables were calculated for every image r , where $r = 1, \dots, 379$. Gray-level images had a total of 17 features used for the classification.

2.3. Supervised classification

The distinctive characteristic of supervised classification techniques is that they contain prior knowledge about the class each element in the set belongs to. In this section we perform supervised classification by applying two different techniques, the Linear Discriminant classifier (LDC) and the K- Nearest Neighbors algorithm (KNN).

The structure of data is given by two possible classes (landscape or non-landscape) and a number of predictor variables (in our work, local variability, effective variance and spatial correlation). These variables were calculated for the two databases previously described in Section 2.1. The distribution of classes in each databases is shown in Table 2.2. The class composed by landscape images is denoted as c_1 and the one

Table 2.2: Classes distribution

Databases	<i>WLW</i>	<i>GLP</i>
Landscape	213	85
Non-landscape	167	294
Total	379	379

composed by non-landscapes is denoted as c_2 . Both procedures are applied by splitting data into two set: a training and testing set. In the training set the class to which each element belongs is known. This set is used to classify the elements of the testing set.

The classification is performed using KNN and LDC techniques and initially using a group of 33 variables for RGB images (and 17 variables for gray-level). We aim to choose the combination of variables which best discriminate both classes. Due to the large number of variables, it became infeasible to conduct the classification for all

possible combinations of them. Thus, we make the classification through a sequential forward selection [also known as wrapping procedures, see [Karegowda et al. \(2010\)](#) and [Kohavi and John \(1997\)](#)]. We choose the *forward* selection in order to include the minimum number of variables in the classification². This technique, widely used in regression problems [see [Hastie et al. \(2009\)](#)], permits to find the subset of variables that minimize the error rate (percentage of misclassified images in the test set). The selection of features is done sequentially [see [Kohavi and John \(1997\)](#), [John et al. \(1994\)](#) and [Karegowda et al. \(2010\)](#)], by adding features to the model one at a time. The procedure begins with an empty feature set and sequentially adds features. The first feature included in the model is the one which individually has the lowest error-rate. The next feature that enters the model is the one that, jointly with the first variable, has the greatest reduction in the error-rate. The process of adding features continues until including a new one does not decrease the error-rate. The resulting subset of features is the ones used to discriminate both classes.

The sequential forward selection technique requires the application of cross-validation method in order to avoid the overfitting of the error-rate. Cross-validation is used to assess how the results of a statistical analysis (in our case, KNN and LDC) will generalize to an independent data set. One round of cross-validation involves the partition of the data into subsets, performing the analysis on one subset (i.e., training set), and validating this analysis on the other subset (i.e., testing set). To reduce variability, multiple rounds of cross-validation (called K-fold cross-validation) are performed using different partitions, and the validation results are averaged over the rounds. Otherwise stated, we divide the data into K roughly equals parts and for each $k = 1, \dots, K$ we apply the classification method using a subset of p features $Z = (Z_1, Z_2, \dots, Z_p)$. Then, we compute the error rate for each k -fold as

$$E_k(Z) = \frac{n_{Ek}}{n_k},$$

where n_{Ek} is the number of misclassified observations (images) in the k th part (fold)

²We also perform the feature selection by backward elimination and some variables were retained unnecessarily making the model less efficient.

and n_k is the size of the k th fold. Hence, the overall cross-validation rate is

$$CV_E(Z) = \frac{1}{K} \sum_{k=1}^K E_k(Z). \quad (2.3.1)$$

The procedure repeats the cross-validation for different subset of features and selects the combination of them that minimize the $CV_E(Z)$. For this work we choose the commonly used 10-fold cross-validation.

To perform the classification we consider five different cases based on possible combinations between different datasets.

1. Classification using the *WLW* database exclusively.
2. Classification using the *GLP* database exclusively.
3. Classification using the *WLW* as training set and *GLP* as testing.
4. Classification using the *GLP* as training set and *WLW* as testing.
5. Classification using both databases as a single set.

For every feature subset constituted by adding a new candidate feature, a 10-fold cross validation is done by repeatedly applying either KNN or LDA, with different training subsamples. In cases 3 and 4, where training and testing are given, the 10-fold cross-validation is performed separating each training and testing set in ten randomly selected subsets. Then, the classification is done considering one of the ten subsets of training with one of the ten subsets of testing set and repeating the procedure 10 trials. In cases 1, 2 and 5 below, with only one database, the whole set is divided into 10 subsets, chosen randomly but with roughly equal size. The classification is done using one of the 10 subsets as the test set and the other 9 subsets (all-together) as the training set. The procedure is repeated 9 more times, and each time a different subset is selected as test set. In all cases the misclassification rate of the candidate subset of feature is calculated as the average error-rate across all 10 trials, as it was explained above.

Linear discriminant analysis

Linear discriminant analysis is a technique used in statistics, pattern recognition and machine learning to find a linear combination of features which separate two or more classes of objects. The purpose is to determine the membership class of a new observation based on a linear combination of variables known as predictors and that belong to a known group.

We have a set of elements (images) which come from two known classes $C = c_1, c_2$ (landscape and non-landscape). In each element a set of p random variables $X = (X_1, X_2, \dots, X_p)$ is observed. The purpose of the linear discriminant classifier is to assign a new element, denoted as ω with known values of the variables X_ω , in one of the two classes. As we conduct the classification assuming unknown populations, the covariance matrices of each class are estimated from the samples.

Let n be the total number of elements, n_c the number of elements in class c , and \bar{X}_c is the vector of mean variables within each class given by

$$\bar{X}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} X_{ic},$$

where X_{ic} is the row vector $1 \times p$ which contains the p values of the variables for the element (image) i in the class c . The classification criterion used in our analysis consists of assigning the element ω to the nearest class using the Mahalanobis distance:

$$D_{mahal_c}^2 = (X_\omega - \bar{X}_c) \hat{S}^{-1} (X_\omega - \bar{X}_c)', \quad (2.3.2)$$

where \hat{S} is the estimated variance-covariance matrix given by

$$\hat{S} = \sum_{c=1}^C \frac{n_c - 1}{n - C} \hat{S}_c \quad (2.3.3)$$

where

$$\hat{S}_c = \frac{1}{n_c - 1} \sum_{i=1}^{n_c} (X_{ic} - \bar{X}_c)(X_{ic} - \bar{X}_c)'. \quad (2.3.4)$$

Finally, the decision rule consists of assigning the element ω to the class c_1 if $D_{mahal_{c_1}}^2 < D_{mahal_{c_2}}^2$, otherwise it is assigned to the class c_2 .

The set of variables used in the classification are those chosen by forward selection for the five cases already explained above. The error-rate and selected features for the five cases are shown in Tables 2.3 and 2.4.

Table 2.3: Linear Discriminant Analysis results for RGB images

Databases	Error-rate	Variables used	Generalization
<i>WLW</i>	3.69%	$EV_R, \rho_{R1}, \rho_{R2}, \rho_{R4}, \rho_{10}^-$ and ρ_{12}^- .	4.88%
<i>GLP</i>	5.14%	$EV_R, EV_G, \bar{\delta}$ and ρ_{1G} .	5.28%
Train= <i>WLW</i> , test= <i>GLP</i>	3.69%	$EV_R, \rho_{R1}, \rho_{R3}$ and $\bar{\rho}_8$.	5.15%
Train= <i>GLP</i> , test= <i>WLW</i>	5.14%	EV_R, EV_G and ρ_{1G1} .	5.20%
Both databases	5.34%	EV_R, EV_G and ρ_{G1} .	

Table 2.4: Linear Discriminant Analysis results for GRAY images

Databases	Error-rate	Variables used	Generalization
<i>WLW</i>	4.61%	δ , and ρ_9 .	5.41%
<i>GLP</i>	8.97%	EV, δ, ρ_1 and ρ_9 .	10.42%
Train= <i>WLW</i> , test= <i>GLP</i>	4.61%	δ , and ρ_9 .	5.01%
Train= <i>GLP</i> , test= <i>WLW</i>	9.23%	$EV, \delta, \rho_1, \rho_2$ and ρ_{10} .	11.21%
Both databases	7.78%	EV and δ .	

Comparing the results of both tables we observe that the error-rates obtained for gray level images are worse than those for RGB images. This differences may be due to in RGB images there are more information to classify than in gray-level images. The results also suggest that *WLW* database lonely has better error-rate in gray and RGB images than *GLP* set. Besides, when using the *WLW* set as training, the classification results are better than when this set is used as testing. This is not surprising because there is a greater heterogeneity of scenes observed in *GLP* database than in *WLW* set. The *GLP* set seems to have, visually, more categories of scenes. For instance, scenes with animals, paintings, people and texture can be found in this set, while these kind of scenes does not appear in *WLW* database.

With respect to the features, it seems to have coincidences among the group of variables selected in each case. The correlations of order 8, 9, 10 and 12 may be included because they give discriminative information about the classes. The structure of the spatial correlation in the datasets shows that there is a group of images with great values of correlation in the first seven orders, decreasing slowly as we increase the order of the variable over to eight. Moreover, there is another group of images where the spatial correlation begins with greater values and decrease quickly as we increase the order of the variable. Therefore, it is reasonable to expect that correlations of orders 8 to 12 have discriminant power between groups.

In order to present a general procedure to classify images with similar characteristics like these databases, we perform the classification considering the variables which are common to all databases. These are: EV_R , EV_G and ρ_{G1} for RGB images, and EV and δ for classifying gray-level images. The last column of Tables 2.3 and 2.4 describe the classification results obtained in each case.

k Nearest Neighbor

The K-nearest-neighbor algorithm is carried out as follows. For each image in the *test* set, the k closest members (nearest neighbors) in the *training* set are found. This proximity is measure by some distance. We chose the cityblock distance (also known as Manhattan) because this measure has provided the best results. The cityblock distance between two variables x_i and y_i is given by

$$d_{cityblock} = \sum_{i=1}^m |x_i - y_i|$$

Then, we obtained the distances of each element in the test set to each member of the training set, and for every test element the k nearest neighbors are recorded. Then, an element is classified to the class most common amongst its k neighbors (majority rule). The procedure is repeated for the remaining cases in the *test* set. The value of k is a positive integer, typically small and even to avoid tie. There is no consensus in defining the number k of nearest neighbors. Often, the value of k is chosen by cross-validation, which is the method that we have used. In this work we consider $k = 3$.

The results of the classification are shown in Tables 2.5 and 2.6. The error-

Table 2.5: K Nearest Neighbor results for RGB images

Databases	Error-rate	Variables used	Generalization
<i>WLW</i>	3.16%	$EV_R, \bar{\delta}$ and ρ_{12} .	4.75%
<i>GLP</i>	5.01%	EV_G, EV_B , and $\bar{\delta}$.	6.60%
Train= <i>WLW</i> , test= <i>GLP</i>	3.43%	$EV_R, \bar{\delta}$ and ρ_{12} .	4.35%
Train= <i>GLP</i> , test= <i>WLW</i>	4.88%	EV_G, EV_B , and $\bar{\delta}$.	6.73%
Both databases	5.47%	EV_R, EV_G and ρ_{G1} .	

Table 2.6: K Nearest Neighbor results for GRAY images

Databases	Error-rate	Variables used	Generalization
<i>WLW</i>	4.61%	EV, δ and ρ_8 .	4.88%
<i>GLP</i>	9.36%	EV, δ, ρ_1 and ρ_5 .	11.48%
Train= <i>WLW</i> , test= <i>GLP</i>	4.61%	EV, δ , and ρ_{13} .	5.41%
Train= <i>GLP</i> , test= <i>WLW</i>	8.97%	EV, δ, ρ_1 and ρ_2 .	9.76%
Both databases	8.70%	EV and δ and ρ_2 .	

rates obtained with KNN are similar that those in LDA and for gray-level images are worse than in RGB images. *WLW* also shows a better performance in classification. Although the selected variables seems to be similar, the mean local variability ($\bar{\delta}$) acquires relevance in some cases of KNN classification. However, in both methods the variables selected when both databases are merged are exactly the same. In order to provide a generalization of the methods, we also obtained the classification rates when the common variables are selected. The last columns of Tables 2.5 and 2.6 contain the results. The variables used in the classification were EV_R, EV_G and ρ_{G1} for RGB images and EV and δ for gray-level images.

Some misclassified images in color-level picture are shown in Figure 2.11. Those images correspond to the merged dataset classified by KNN algorithm. In (a) are examples of pictures wrongly classified as landscapes, and in (b) are examples of those

wrongly classified as non-landscape. As it is observed, the images wrongly classified in

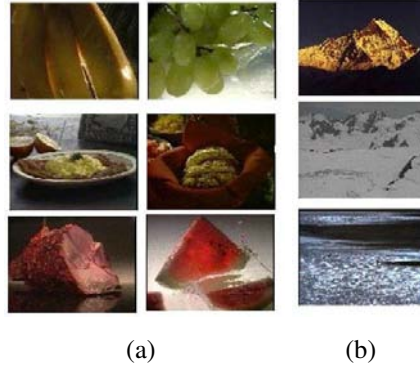


Figure 2.11: Examples of misclassified color-level pictures- Merged set

landscape group seems to have homogeneity in colors, one of the typical characteristic of this group.

2.4. Comparison to Support Vector Machine

In order to give an idea of the performance of the classification procedure proposed in this chapter, we apply support vector machine statistical classifier (SVM) for binary linear classification. In the support vector machines [see [Boser and et al. \(1992\)](#) and [Cortes and Vapnik \(1995\)](#)], a data point is viewed as a p -dimensional vector and the goal is to know whether such points can be separated with an hyperplane. Specifically, in image analysis context, the p -dimensional vector is obtain as follows. Given an image x_i of size $n \times m$, the information of each image is stored in a p -dimensional vector formed by the concatenation of its column (or rows) pixels. Then, p is the product of n and m , $p = n \times m$. Then, $x_i \in \mathbb{R}^p$.

There are many hyperplanes that might classify the data. One reasonable choice is to consider as the best hyperplane the one that represents the largest separation, or margin, between the two classes. Then, it is chosen the hyperplane that maximizes the distance from it to the nearest data point on each side. If such hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier defined is known

as a *maximum margin classifier*. In other words, it is given a training data T composed by a set of n points of the form

$$T = \{(x_i, y_i) / x_i \in \mathbb{R}^p, y_i \in \{1, -1\}\}_{i=1}^n \quad (2.4.1)$$

where the y_i is either 1 or -1 , indicating the class to which the point x_i belongs and each x_i is a p -dimensional vector. The goal is to find the maximum-margin hyperplane that divides the points having $y_i = 1$ from those having $y_i = -1$. Any hyperplane can be written as the set of points \mathbf{x} satisfying

$$\mathbf{w}^T \mathbf{x} + b = 0, \quad (2.4.2)$$

where $\mathbf{w} \in \mathbb{R}^p$ is a vector perpendicular to the hyperplane, and b is a constant that determines the position. The values of \mathbf{w} and b have to be chosen to maximize the margin or distance between the parallel hyperplanes, that are as far apart as possible. We are looking for an hyperplane such that by the projection of all observations of the class 1, we obtain

$$\mathbf{w}^T x_i + b \geq 1, \quad (2.4.3)$$

and for class -1 , we obtain

$$\mathbf{w}^T x_i + b \leq -1. \quad (2.4.4)$$

This can be rewritten as,

$$y_i(\mathbf{w}^T x_i + b) \geq 1 \quad (2.4.5)$$

Moreover, the distance from any points to the hyperplane is obtained as

$$d(x_i; \mathbf{w}, b) = \frac{|\mathbf{w}^T x_i + b|}{\|\mathbf{w}\|}. \quad (2.4.6)$$

The maximum margin classifier looks for the hyperplane that maximizes the distance between two sets of points of two parallel hyperplanes (called support vectors, see Figure 2.12). Note that if the training data are linearly separable, we can select the two hyperplanes of the margin in a way that there are no points between them, and then try to maximize their distance. Therefore, we find that the distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$, and the maximization of the margin is reduced to minimizing the norm $\|\mathbf{w}\|$ of the vector that is perpendicular to the optimum hyperplane.

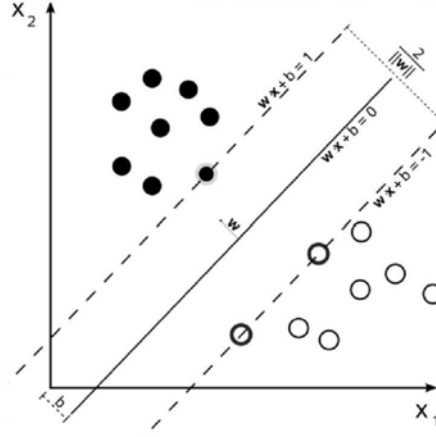


Figure 2.12: Maximum-margin hyperplane and margins for an SVM trained with data from two classes. Points on the margin are called the support vectors.

In order to consider classes that are not linearly separable, it is introduced a slack variable denotes as ξ_i which measure the degree of misclassification of the datum x_i . Thus, the optimum hyperplane is obtained solving the optimization problem given by

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i, \quad (2.4.7)$$

subject to

$$y_i(\mathbf{w}^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (2.4.8)$$

where $\xi = \{\xi_1, \dots, \xi_n\}$ is the vector of slack variables and C is the penalty parameter of the error term. The solution of the constraint minimization problem of equation 2.4.7 can be solved using Lagrange multipliers.

The original optimal hyperplane algorithm proposed by [Vapnik and Lerner \(1963\)](#) was a linear classifier. However, [Boser and et al. \(1992\)](#) suggested a way to create non-linear classifiers by applying a kernel to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that the product $\mathbf{w}\mathbf{x}$ is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed

space high dimensional. Then, the optimization problem is given by,

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i \quad (2.4.9)$$

subject to

$$y_i(\mathbf{w}^T \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad (2.4.10)$$

where training vectors x_i are mapped into a higher dimensional space by the function ϕ . SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. Furthermore, $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is called the kernel function. Finally, the decision rule for a new point \mathbf{z} is given by the decision function

$$f(\mathbf{z}) = \{\mathbf{z}^T \mathbf{w} + b\} \quad (2.4.11)$$

where \mathbf{w} are the support vectors, and the sign of $f(\mathbf{z})$ is negative if \mathbf{z} belongs to class -1 and positive if belongs to class 1 .

For the application conducted in this Chapter we consider the image in RGB color. Accordingly, each image is represented as a row-vector composed with the concatenated rows of the three RGB matrices. For instance, an image of size $n \times m = 80 \times 120$ gives an input row vector of $1 \times (n \times m \times 3) = 1 \times 28800$. The procedure is conducted applying 10-fold cross-validation. In each round of this cross validation we performed another cross-validation to find the adequate penalty parameter C (see equation 2.4.9). We use two different kernels given by

- Linear

$$K(x_i, x_j) = x_i^T \cdot x_j. \quad (2.4.12)$$

- Radial basis function (RBF)

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \quad \gamma > 0. \quad (2.4.13)$$

where γ is a kernel parameter [see [Buhmann \(2003\)](#) for more information]. The parameter gamma is also selected by cross-validation in the same way as we calculate the parameter C .

As result, we achieve a mean error-rate of 12,46% obtained for the 10-fold as it is explained in equation 2.3.1. Comparing this outcome with those from our methods, we conclude that the variables and procedures proposed in this thesis are significant superior in terms of classification rates. While in SVM the image is represented by 28800 variables, in our method an image is represented by only three low level features. Consequently, our method is less time-consuming in terms of processing.

2.5. Unsupervised classification

The unsupervised classification is performed to confirm the fact that there are two sets of images in the databases. We apply the cluster technique based on the *k-means* algorithm, more appropriate for high dimensional data. The K-means algorithm requires the number of clusters fixed a priori. The main idea of this algorithm is to define k initial class centers, one for each cluster. In general, the class centers are randomly selected or chosen as much as possible far away from each other. In each iteration process, elements are assigned to the nearest class, and new class centers are calculated. The new class centers are the points that minimizes the sum of the squared distances between points in the class and the respective class center. In each new iteration, class centers shift and the class assignments for some elements may change. The process is repeated until some optimality criterion (previously defined) is achieved.

In this section we aim to form two groups and observe how the sum of squares within groups decreases when going from one to two. In order to observe these changes, we carry out the *F test of variability reduction*, frequently used to calculate the relative reduction of variability with the increase of an additional group. The test compares the sum of squares within groups with k and k+1 groups. Specifically, we compared the variability of having one group ($k = 1$) with that one obtained with two groups ($k = 2$). Then, the F-test is given by

$$F = \frac{SSW(K) - SSW(K+1)}{\frac{SSW(K+1)}{n-K-1}},$$

where n is the sample size. The SSW is the sum of squares within groups obtained as

$$SSW = \sum_{k=1}^K \sum_{j=1}^p \sum_{i=1}^{n_k} (x_{ijk} - \bar{x}_{jk})^2, \quad (2.5.1)$$

where x_{ijk} is the value of j -th variable, in the i -th element of k -th group, and \bar{x}_{jk} is the mean of the j -th variable in the group k . This F value is compared with a value obtained using an F distribution with p , and $p(n - K - 1)$ degrees of freedom, where p is the number of variables used in the grouping procedure.

We conducted the K-means algorithm using the set of variables selected in the case 5 of the previous section for RGB images . These variables are, EV_R , EV_G and ρ_{G1} . The results are shown in Tables 2.7 and 2.8.

Table 2.7: K-MEANS algorithm information- *WLW* database

Cluster number	Sum of Squares within groups	F-statistics
1	4.40	484.31
2	1.93	

Table 2.8: K-MEANS algorithm information- *GLP* database

Cluster number	Sum of Squares within groups	F-statistics
1	10.579	315.93
2	5.84	

The outcomes suggest that in both databases two groups can be formed, because the variability reduction is significant when going from one to two. Besides, in *WLW* database the variability is lower. For instance, when two groups are formed, *WLW* has a $SSW = 1.93$, while in *GLP* the $SSW = 5.84$. This behavior is coherent with the results obtained in previous section which the classification shows better error-rates for *WLW* dataset.

2.6. Conclusions and contributions

The classification of images by scenes is a complex process, since sometimes the differences among image scenes are not obvious, even for human vision. For that reason, this kind of classification is influenced by the subjectivity of the human thought. Then, it becomes difficult the generalization of the techniques for the classification of any scene.

In this chapter we propose three low-level features obtained by considering the variability and dependency of pixels in the images. The proposed variables, through different directions, capture the contrast of color intensities observed in images. This contrast helps to differentiate the group of landscape and non-landscape scenes because, for instance, pixels composing an image of a sky are homogeneous with respect to the heterogeneity of pixels in a Miró's painting image. In addition, while the effective variance and spatial correlation show differences in the values obtained for the three RGB matrices, the local variability do not show differences. Some variables calculated initially are not considered in the classification through the sequential forward selection. For example, the spatial correlations of first orders such as $h = 1, 2$ and middle orders such as $h = 7, 8$ and 9 seem to be more powerful to discriminate both groups than other orders of correlations. This greater discriminative power can be explained by the structures of correlation commonly observed in our databases, where these orders of correlations show differences in both groups.

The classification rates obtained for color-level images are better than results obtained for gray-level images. This behavior seems to be as a consequence of the greater amount of information contained in the three RGB matrices than that in the gray-level matrix. The classification rates achieved by the K-nearest neighbors and the linear discriminant techniques do not seem to be significantly different. However, those classifiers report a good performance having better results than the ones obtained by support vector machine techniques over the same databases. Besides, our classification rates improve results achieved by other authors in this kind of classification.

2.7. Future work

There are several lines of research arising from this chapter which should be pursued. Firstly, the application of the procedures to a different database with similar characteristics, in order to observe the behavior of the proposed variables in classifying other kind of scenes (e.g., texture vs. non-texture or indoor vs. outdoor scenes).

Secondly, we aim to study in depth the variation observed in the effective variance when there is changes in matrix dimensionality. It is our goal to find the statistical explanation of this behavior.

Finally, a line of investigation already in progress is the study of the spectral function obtained from the image. This is another possible classification measure which uses all the correlations jointly. Some exploration conducted about this function is shown following.

Intuitively, the spectral density (also known as power spectral density) captures the frequency content a stochastic process and helps identify periodicity, for instance in image's rows. The spectral representation is related with the idea that a time series is composed by periodic components, appearing in proportion to the underlying variances.

We analyze a digital image as a regularly spaced series of values, where the values are the intensities red, green or blue of pixels. Each row (or column) can be seen as a discrete time series or as a discrete values succession that can be transformed to the frequency domain in order to study the rates of oscillation or frequencies of them. Assuming that each row (or column) is an stationary process (i.e., constant variance), the autocovariance function $\gamma(h)$ satisfying

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty$$

has the representation

$$\gamma(h) = \int_{-1/2}^{1/2} e^{2\pi i v h} f(v) dv \quad (2.7.1)$$

for $h = 0, \pm 1, \pm 2$, where $f(v)$ is the spectral density function and v is the frequency index, defined in cycles per unit time. That is, for $v = 1$ the series makes one cycle

per time unit; for $\nu = 0.5$, the series make a cycle every two time units; for $\nu = 0.25$ every four units, and so on. Normally, data that occurs at discrete time points, as in this case, will need at least two points to determine a cycle. Then, the highest frequency of interest is $\nu = 0.5$ per point. The spectral density is obtained as the inverse transform of the autocovariance function as

$$f(\nu) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \nu h} \quad (2.7.2)$$

for $-1/2 \leq \nu \leq 1/2$. Since $f(\nu)$ is symmetric with respect to ν_0 , the range of variation of this variables can be changed to $0 \leq \nu \leq 1/2$ [see [Shumway and Stoffer \(2010\)](#)]. Hence, the power spectral density (PSD) function in 2.7.2 can be rewritten as

$$f(\nu) = \sum_{h=0}^{\infty} \gamma(h) [2 \cos(2\pi \nu_0 h)] \quad (2.7.3)$$

One way to estimate the spectral density function is through the analysis of the periodogram. The periodogram is obtained as a sample version of the spectral density function in expression 2.7.2,

$$P(\nu_j) = \frac{4}{n} \sum_{h=-(n-1)}^{n-1} \hat{\gamma}(h) [2 \cos(2\pi \nu_0 h)] \quad (2.7.4)$$

However, the raw periodogram is not a good spectral estimate because of spectral bias and the fact that the variance at a given frequency does not decrease as the number of samples increases. The spectral bias problem can be reduced multiplying the finite sequence by a window function which truncates the sequence gradually rather than abruptly. The variance problem can be reduced by smoothing the periodogram. One of the techniques commonly used to solve the variance problems is to apply a window. We apply the Blackman and the Bartlett windows to smooth the periodogram. The Blackman window is based on Fourier transformation of the smoothed, truncated autocovariance function and is defined as

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right) \quad (2.7.5)$$

where $a_0 = \frac{1-\alpha}{2}$; $a_1 = \frac{1}{2}$; $a_2 = \frac{\alpha}{2}$ and $\alpha = 0.16$.

The Bartlett window with zero-valued end-points is obtained as

$$w(n) = \frac{2}{(N-1)} \left(\frac{N-1}{2} - \left| n - \frac{N-1}{2} \right| \right) \quad (2.7.6)$$

where N represents the width, in the sample, of a discrete-time window function and n is an integer with values $0 \leq n \leq N-1$.

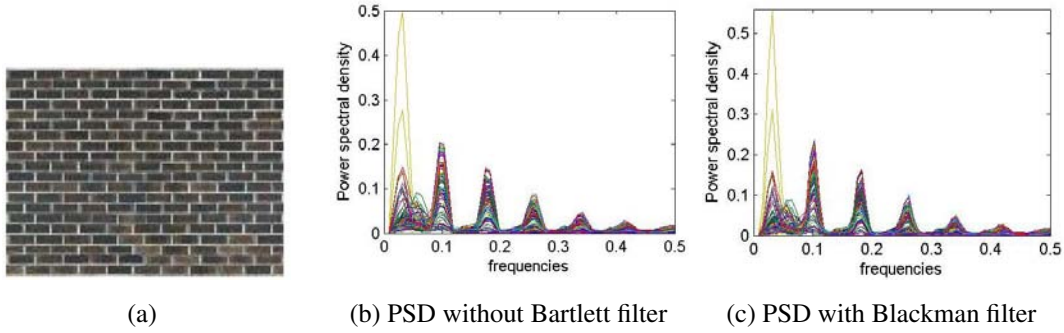


Figure 2.13: Example of Periodogram for a texture image

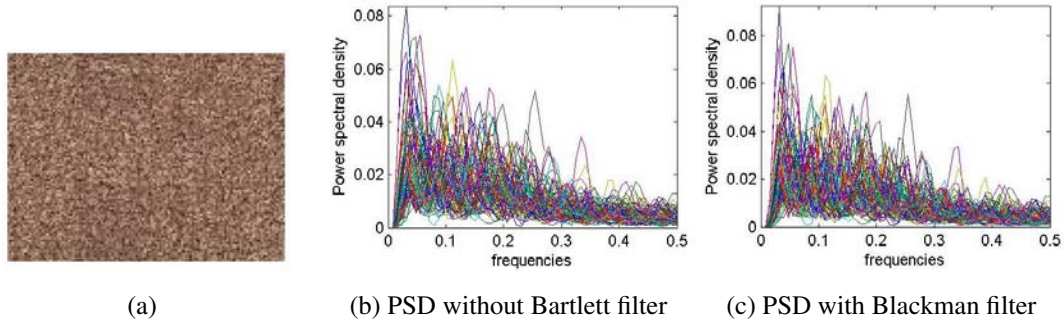


Figure 2.14: Example of Periodogram for a texture image

Graphically, Figures 2.14, 2.13 and 2.15 show the periodograms of two images of textures and a picture of a tree. The x-axis represents the frequency ν and the y-axis the power spectral density function. In every periodogram we represent the PSD of each row of the image. The periodograms obtained with both windows are very similar. Comparing the figure plots, it is observed that in Figure 2.13 the rows of the image show cycles with similar frequencies, whereas in Figure 2.14 do not. Moreover,

in Figure 2.15 all rows seem to have cycles with the same frequencies. This behavior could be information about different groups such as textures.

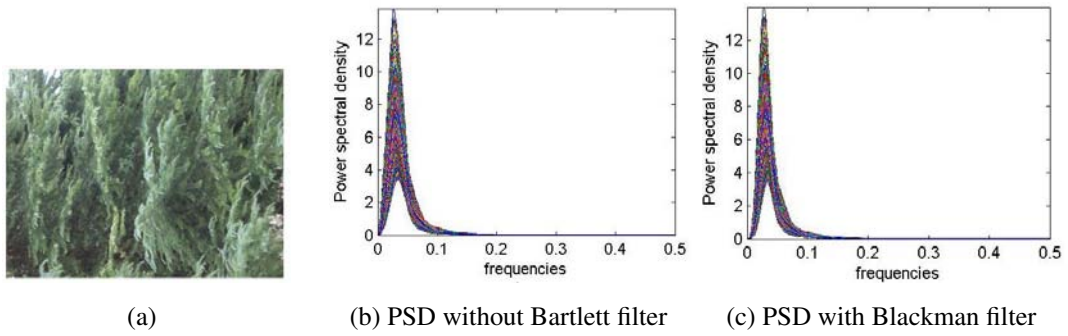


Figure 2.15: Example of Periodogram for a scene image

With this briefly exploration we believe that the spectral analysis represents an issue in itself and probably deserves a separate study of its own. As such, we suggest it as a promising avenue for future research.

CHAPTER 3

Handwritten Digit Classification

In the field of digital image processing, pattern recognition is the research area that studies the operation and design of systems that recognize patterns in data. Its primary goal is the classification of objects into a number of categories or classes. Depending on the use, these objects can be images, signal waveforms or any other type of measurements that need to be classified. Common applications of pattern recognition are automatic speech recognition, classification of text into several categories (e.g. spam/non-spam email messages) and the automatic recognition of handwritten postal codes on postal envelopes. Statistical approaches are one of the most widely studied and used to perform pattern classification.

In statistical pattern recognition, an image is represented by a set of f features which constitute a f -dimensional feature vector. A decision process (statistical classifier) based on this vector is used to establish boundaries between image classes and then perform the classification. Thus, the classification success depends entirely on the set of selected features and the classification scheme.

In this chapter we tackle the handwritten digit recognition problem. Our purpose is to present alternative classification methods based on statistical techniques and with

good performance for classifying handwritten digit images. We use two different statistical approaches. Firstly, we use a probabilistic approach assuming that the features of images in the training set have a probability distribution in order to use the Bayes's decision rule to classify images in the test set. Secondly, we conduct a supervised classification approach (where classes of sets are known), using the K-nearest neighbors rule to classify images in the test set. The classification scheme is based on a set of variables (feature vector) obtained by applying structural measures to detect the shape and geometry of the numbers. Our methodology has the advantage that is more intuitive and generalizable over other methods that require the use of scanned digit with the same size [see [Lauer et al. \(2007\)](#)]. Besides, our methodology do not need any pre-processing (as deskew, noise removal or shift the edges) of images from databases [see [Decoste and Scholkopf \(2002\)](#) and [Keysers et al. \(2007\)](#)].

Experiments are performed on two databases described in Section 3.1. The binarization of the images is required to calculate the features used in this chapter. Thus, in Section 3.2 we propose a new binarization method to find an optimum threshold parameter for each image, which is based on the written trace of digit. Section 3.3 describes the construction of the variables proposed for the classification. Initially, the feature vector is composed with the variables calculated in Section 3.3. However, the final feature vector used to classify is selected by the application of the sequential forward selection technique, already explained in Section 2.3 in the Chapter 2. In Section 3.4 we present the probabilistic approach which is performed by the application of the Bayes's rule, disjointing the variables into categorical and quantitative. Section 3.5 is devoted to describe the application of the K-nearest method algorithm to classify the digit images. Finally, the last Section 3.6 provides the conclusion of the two classification methods. Both techniques provide similar results to handwritten digit classification, despite the fact that in the probabilistic approach we consider the distribution of the variables, whereas in the KNN algorithm only the distances are used in the classification.

3.1. Databases

The classification in this chapter is performed using two different databases consisting of scanned digits, from 0 to 9. Datasets are partitioned in training and testing sets in order to validate the technique. In general, as explained in Chapter 2, training set is used to classify the testing set, being both independent.

MNIST database

The *MNIST* database was constructed from National Institute of Standards and Technology (*NIST*) database of scanned handwritten digit.

NIST originally had the training set composed by a collection of digits written by paid US census workers, while the testing set was collected from digits written by uncooperative high-school students. This difference of origin of the data explains why the classification errors obtained in each group were completely different with worse performance on the test data. Therefore, the *NIST* database required a reorganization in order to combine adequately training and test sets, forming the *MNIST* (Modified *NIST*) set [LeCun et al. (1998)], which is the database that we use in this dissertation.

The *MNIST* database is composed by 60.000 handwritten digits in the training set and 10.000 in the test set. Digits were size-normalized and centered in an image of size 28×28 by computing the center of mass of pixels, and translating the image to locate this center point at the center of the 28×28 field [LeCun et al. (1998)]. The *MNIST* set contains gray level images as a result of the anti-aliasing technique used by the normalization algorithm conducted by the authors (<http://yann.lecun.com/exdb/mnist/>). The distribution of digits in the training and testing sets are in Table 3.1.

Table 3.1: Distributions of *MNIST* sets

Digit	0	1	2	3	4	5	6	7	8	9	Total
Train	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949	60000
Test	980	1135	1032	1010	982	892	958	1028	974	1009	10000

Some examples of *MNIST* are in Figure 3.1. As a consequence of the origin of the

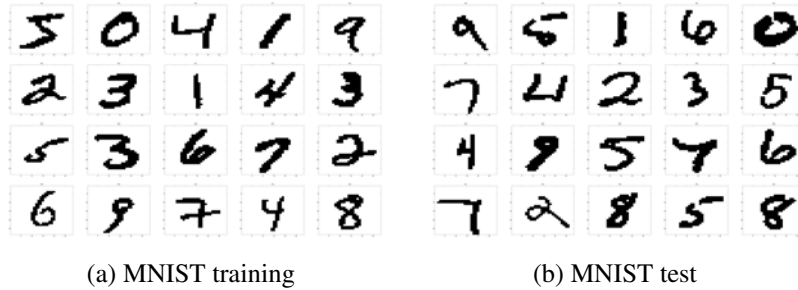


Figure 3.1: Typical Images from *MNIST* sets

MNIST database, digits in the training and testing sets seem to have the same degree of difficulty to be recognized.

USPS database

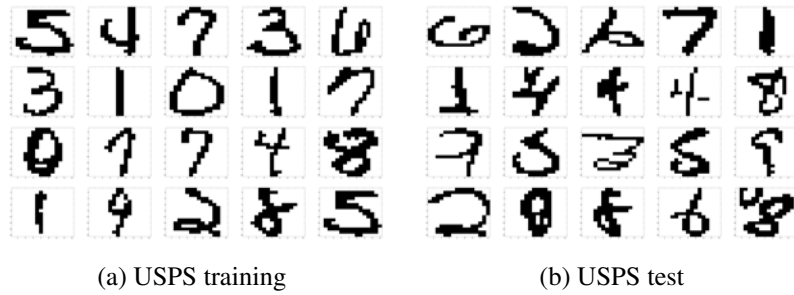
The *USPS* database comes from a set of digits automatically scanned from envelopes by the United State Postal Service. The original scanned digits were binary and with different sizes and orientations. The segmentation procedure performed by Postal Service caused that some digits were mis-segmented. Thus, the database was generated with extreme difficulty to be recognized and classify, with a human error-rate around the 2,5% [Simard et al. (1993)]. Images used in this work were deslanted¹ and size-normalized by LeCun et al. (1990), resulting in 16 x 16 grayscale images. The training set is composed by 7291 images and the testing set contains 2007 images. The distributions of digits in the training and testing sets is showed in Table 3.2. Examples of this database are in Figure 3.2.

Although the completely set of numbers is not shown, it can be seen in the examples of *USPS* presented in Figure 3.2 some clear differences between test and training sets. In the testing set digits seem to be more unreadable than in the training set. Besides, the testing has some mis-segmented digits, such as the digit 8 shown in Figure 3.2b. This

¹Deslant is a term commonly used in handwritten recognition field to indicate the action of removing the slant of the text by some specific technique.

Table 3.2: Distributions of USPS sets

Digit	0	1	2	3	4	5	6	7	8	9	Total
Train	1194	1005	731	658	652	556	664	645	542	644	7291
Test	359	264	198	166	200	160	170	147	166	177	2007

Figure 3.2: Typical Images from *USPS* sets

obvious discrepancy between sets represents a difficulty to classify both sets with the same performance in terms of classification, because the training, set used to classify images of the testing set, has less variability in the shape of digit.

3.2. *Binarization*

The variables proposed in this chapter to make handwritten digit classification require images in binary level. The binarization process assumes that images contain two classes of pixel: the foreground (or white pixels, with maximum intensity, i.e., equal to 1) and the background (or black pixels with minimum intensity, i.e., equal to 0). The goal of the method is to classify all pixels with values above of the given threshold as white, and all other pixels as black. That is, given a threshold value t and an image X with pixels denoted as $x(i, j)$, the binarized image X_b with elements $x_b(i, j)$ is obtained as follows.

$$\begin{aligned} \text{If } & x(i, j) > t, & x_b(i, j) &= 1 \text{ (object)} \\ \text{else} & & x_b(i, j) &= 0 \text{ (background)} \end{aligned}$$

However, this version of the algorithm assumes that we are interested in light objects on a dark background. Then, in order to obtain dark objects on a light background we would use,

$$\begin{aligned} \text{If } x(i, j) < t, \quad x_b(i, j) &= 1 \text{ (object)} \\ \text{else} \quad x_b(i, j) &= 0 \text{ (background)} \end{aligned}$$

Then, the key problem in the binarization is how to select the correct threshold t for a given image. We observe that the shape of any object in the image is sensitive to variations in the threshold value, and even more sensitive in the case of handwritten digit. Examples of digit binarized with different threshold value are shown in Figure 3.3. In the upper panel there is an example of gray level digits that appear binarized by different threshold values in the middle and lower panels. A simple look to the lower panel detects that some digits are partially missed by the binarization process. In some cases, the trace of the digit line is cropped, making their recognition more difficult. Therefore, we consider that a binary handwritten number is better recognized

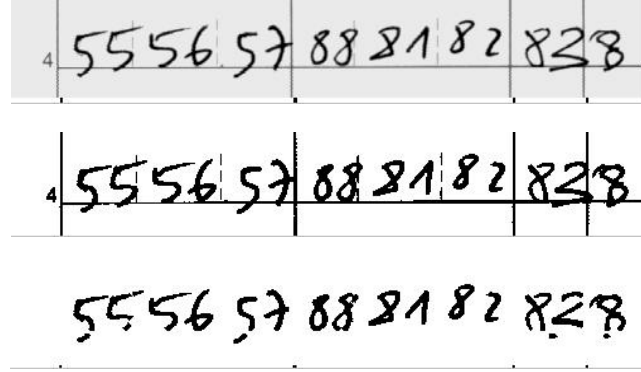


Figure 3.3: Examples of binarized digits

computationally if its trace is complete and continuous, this is the criterion that we use to the threshold, being its choice of crucial importance.

Although several methods exist for choosing a threshold, in an attempt to obtain a threshold more adequate for digit binarization, we propose a novel method to find an optimum threshold value. The procedure is based on statistical concepts that consider the handwriting trace of the digit, finding an optimum threshold value associated with

each image. In the process we use the median absolute deviation² (mad) as a robust variability measure. The algorithm to find the optimum threshold consists of assigning an initial threshold value denoted as t and then binarize the image X , obtaining what we call the *local trace of the line* in a digit. That is, for each white pixel x_{ijt} (pixel binarized with value 1, located in row i , column j and with a given threshold t), we find the horizontal and vertical number of contiguous white pixels, denoted as h_{ijt} and v_{ijt} respectively. Then, we choose the minimum of both values, indicated as Y_{ijt} . That is, Y_{ijt} is the trace of the line at point (i, j) with threshold t . Thus, we define the median absolute deviation of the values Y_{ijt} as the *global variability of the trace of the line*, denoted as $gtr(X)_t$. The procedure is repeated for different threshold values and the optimum threshold (t_{op}) for an image X is the value of t that has the minimum *global variability of the trace of the line*, that is, the value that makes the trace more homogeneous. Formally,

$$Y_{ijt} = \min(h_{ijt}, v_{ijt}) \quad (3.2.1)$$

$$gtr(X)_t = mad(Y_{ijt}) \quad (3.2.2)$$

$$t_{op} = \min_t(gtr(X)_t) \quad (3.2.3)$$

Figure 3.4 shows the same digit with two different threshold values. In both cases the digit is clear to be recognized as the number three by human eye. However, both images are not clear enough for a computer to be classified in the same class. The digit with a $t = 0.001$ is more difficult to recognize than the one that has $t_{op} = 0.30$.

3.3. Features extraction

The group of features used in the classification scheme considers the shape and some structural characteristics of the digits. Some of them are calculated through the

²In general, the mad of a variable X is defined as $mad(X) = median_i(|X_i - median_j(X_j)|)$, for $i = 1, \dots, n$ and $j = 1, \dots, n$.

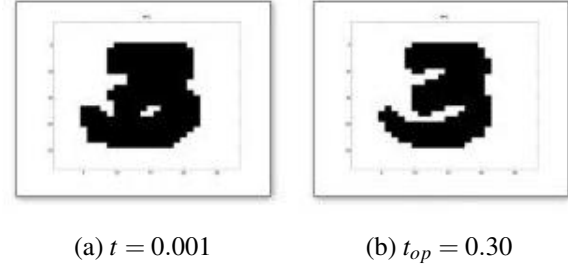


Figure 3.4: The same digit with different threshold values

Hough transform and others are proposed, developed and programmed specifically for this work, using the software MATLAB version 7.9 [see [Gonzalez et al. \(2004\)](#)]. All the variables described in this section are obtained for every digit binarized with its corresponding optimum threshold value described in the previous section. Since some of the features are based on the Hough Transformation, we explain it in detail in the next subsection.

3.3.1. Hough Transform

The Hough Transformation is a technique initially implemented to the identification of lines in a binarized image. Later, it was extended to detect arbitrary shapes, generally circles or ellipses [[Ballard \(1981\)](#)]. The HT, as it is universally used today was developed by [Duda and Hart \(1972\)](#). However, the name comes from its inventor [Hough \(1962\)](#). In this dissertation we use the HT to detect lines and circles in digit images.

Straight lines detection

The main tenet of the HT is to detect the occurrence of figure points (pixels for us) in an image, lying on a straight line. The equation for a straight line is represented in the Cartesian coordinates as

$$y_i = ax_i + b, \quad (3.3.1)$$

and is graphically plotted for a pair of points (x_1, y_1) and (x_2, y_2) as in Figure 3.5a. The HT aims to find points with coordinates (x, y) that satisfy the equation 3.3.1.

There exist infinite lines which pass through a particular point (x_i, y_i) in the Cartesian plane, but only one line satisfies the equation 3.3.1 for specific values of parameters a and b . Moreover, points lying on the same straight line in Cartesian plane can be represented in the space of parameters a and b as it is shown in Figure 3.5b. That

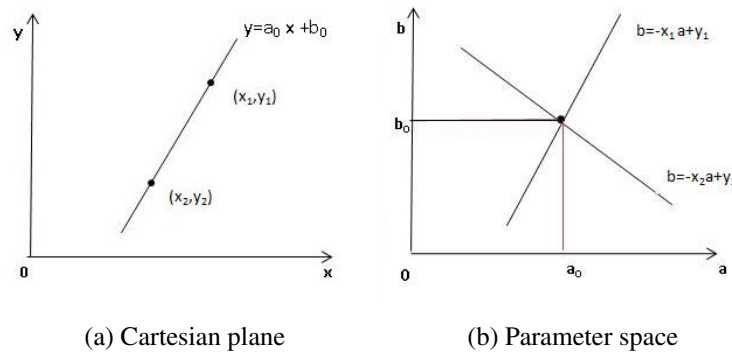


Figure 3.5: Representation of a line

means, two points lying on the same straight line with parameters a and b in the Cartesian plane are represented in the parameter space as two lines with an interception point (a, b) . Then, an arbitrary straight line can be represented by a single point in the parameter space.

A disadvantage of using the equation 3.3.1 is that the slope can be infinite if the straight line is vertical. This problem is solved by using the so-called normal representation of a straight line (also known as Hesse's normal).

The general form of the linear equation 3.3.1 is

$$Ax + By + C = 0, \quad (3.3.2)$$

and the normal representation is given by

$$x \cos \theta + y \sin \theta - \rho = 0. \quad (3.3.3)$$

As both form represent the same line, their respected coefficients must be proportional.

Therefore,

$$\begin{aligned}\cos \theta &= k A \\ \sin \theta &= k B \\ -\rho &= k C,\end{aligned}\tag{3.3.4}$$

where k is a coefficient of proportionality. Squaring and summing both sides of the first and the second equation in 3.3.4, we obtain

$$\cos^2 \theta + \sin^2 \theta = k^2(A^2 + B^2).\tag{3.3.5}$$

Hence,

$$k = \frac{1}{\pm\sqrt{A^2 + B^2}}.\tag{3.3.6}$$

Substituting in equation 3.3.4,

$$\begin{aligned}\cos \theta &= \frac{A}{\pm\sqrt{A^2 + B^2}}, \\ \sin \theta &= \frac{B}{\pm\sqrt{A^2 + B^2}}, \\ -\rho &= \frac{-C}{\pm\sqrt{A^2 + B^2}},\end{aligned}\tag{3.3.7}$$

which are the coefficients of equation 3.3.3. In our case, $A = a$, $B = -1$ and $C = b$.

An example of a line (called ℓ) obtained by the equation 3.3.1, with parameters a_0 and b_0 is represented in red color in Figure 3.6a. The parameter ρ is the vector that

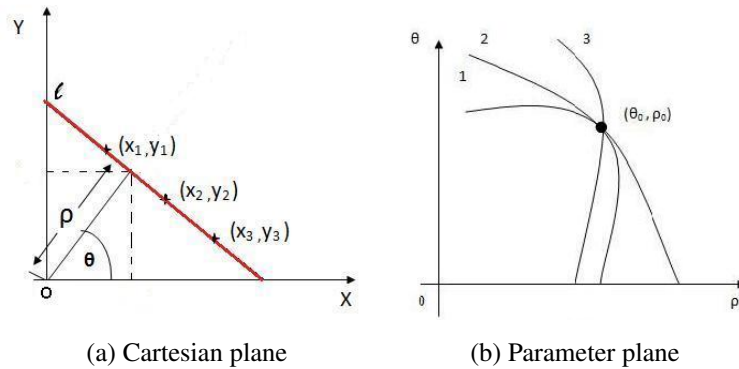


Figure 3.6: Normal representation of a line.

represents the distance between the line ℓ and the origin, while θ is the angle that forms the vector ρ with the X-axis.

As a consequence of this representation, each point (x,y) in the normal form (Figure 3.6a) is represented as a curve in the parameter plane (Figure 3.6b). If we add the restriction that θ belongs to the interval $[0, \pi]$, the normal parameters for a line are unique. With this restriction, every line in the cartesian plane corresponds to a unique point in the parameter plane. The collinear points located in a straight line in normal representation have a common point of intersection in the parameter plane. The point (θ_0, ρ_0) in this plane defines the line passing through the collinear points in the Cartesian plane. Thus, the problem of detecting collinear points can be converted into a problem of finding a common point of intersection.

The Hough transform algorithm uses an array (or matrix) called accumulator, to detect the existence of straight lines in images. The dimension of the accumulator is given by the number of unknown parameters in the equation 3.3.3, i.e. two. The parameters are quantized to be represented in a two-dimensional array with size $d_1 \times d_2$, where d_1 is the number of values of θ uniformly spaced in the interval $(0, \pi)$, and d_2 is the number of values in the ρ axis also uniformly spaced in an interval specified as $(-R, R)$. Then, for each pixel of the image, the accumulator eventually records the total number of lines (with restricted parameters), passing through the pixel (see Figure 3.7). After all pixels are treated, the array is inspected to find cells with high counts,

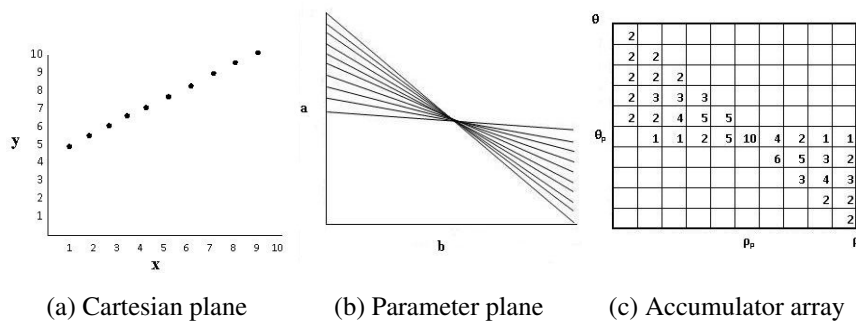


Figure 3.7: Hough Transform

called peaks (p). If the counts of a given cell is 10, then, there are precisely 10 points

lying along the line whose normal parameters are (θ_p, ρ_p) . In Figure 3.7c there is an example of the accumulator array (or Hough matrix) where there is a peak $p = 10$. Thus, there is a line in the image composed by 10 points (or pixels) represented as in Figure 3.7a. In Figure 3.7b, the 10 lines are represented in the space of parameter a and b . Finally, in Figure 3.7c is shown the resulting accumulator array of the Hough transform. The matrix shows the lines that can be formed in a picture, where each cell represents the number of pixels composing them. The longest line is the one which has 10 points (pixels). However, there is one line with 6 points and four other lines composed by 5 points. The lines to be selected depend on the goal of the study.

In the HT calculation, the parameters ρ and θ can be restricted to find lines with a particular slope or position in an image. Also the minimum number of pixels required to conform a line (the minimum value of a peak p) can be determined. After making some analysis in our work, we consider interesting to find vertical (90°), horizontal (0°) and diagonal (45°) lines to differentiate digits. Since images have small size we select lines with at least two pixels. Therefore, the Hough transform detected all possible lines with those characteristics in four stages.

1. In the first stage the binary image is split horizontally into two rectangular equal parts and the largest horizontal line is registered from each part of the image.
2. In the second stage, the binary image is divided vertically into two rectangular equal parts and the largest vertical line is detected from each part of the image.
3. In the third stage, the image is divided by its principal diagonal and the largest upper and lower parallel to this diagonal are found.
4. In the four stage, the image is divided by its secondary diagonal and the largest upper and lower parallel to this diagonal are found.

The features considered in the classification were obtained from the information of the selected lines. Every line has two points, the start and the end-point. The coordinates (x_i, y_i) which specify the start-point and end-point of a straight line are what we call *straight* (S) to specify the coordinate of lines of 0° and 90° . Besides,

we called *diagonal* (D) to refer the coordinates of a diagonal lines (with 45°). We also include the *length* (Le) of each line as an additional variable. Therefore, we have a total of 5 variables for every line (4 corresponding to the coordinates and 1 corresponding to the length). Since we use four straight lines and four diagonals to defined the shape of an image, we record 40 resulting values per image. Graphically, examples of straight and diagonal lines are depicted in Figure 3.8.

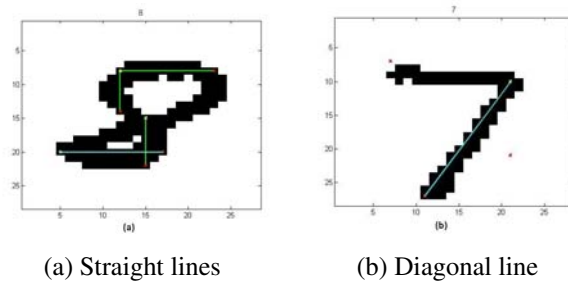


Figure 3.8: Examples of lines

During the calculation of the Hough transform, we observe that some digits do not have all the lines we want to extract, that is 2 horizontal, 2 vertical, and 4 diagonal lines to define the shape of the digit. In these cases, we replace the missing information by assigning the coordinates and length of a single point. The coordinates of those points are determined by the location of them in the image. Each replacement point is located in accordance with the line that is missed. The replacements points for each missing line are shown in Figure 3.3.







Circles detection

We also used the HT to detect circles in images. In an xy Cartesian plane (see Figure 3.9), the circle with center coordinates (a, b) and radius r is the set of all points (x, y) such that

$$(x - a)^2 + (y - b)^2 = r^2 \quad (3.3.8)$$

For a circle with radius r and center in the origin $(0, 0)$, equation 3.3.8 is rewritten

Table 3.3: Missing straight lines

missing line	graphically	missing line	graphically
horizontal up		vertical left	
horizontal down		vertical right	
principal diagonal		secondary diagonal	

as,

$$x^2 + y^2 = r^2.$$

Then, the coordinates (x, y) in the Cartesian plane center in this origin are equal to

$$x = r \cos \phi$$

$$y = r \sin \phi,$$

where ϕ is the angle that the radius forms with the x -axis, defined in a range $(0, 2\pi)$.

And, by axis translation, those coordinates center in (a, b) are equal to

$$x = r \cos \phi - a$$

$$y = r \sin \phi - b.$$

Resulting that

$$a = r \cos \phi - x$$

$$b = r \sin \phi - y.$$

The Hough transform finds all circles with an specific radius r , with an angle θ and centered in the point (a, b) (see Figure 3.10). The accumulator array in this case has 3 dimensions given by the three parameters of equation 3.3.8. Then, each cell of the accumulator array gives the number of pixels lying in a circle with parameters a , b and r . The cell containing a peak represents the circle with highest number of pixels in the image. However, the existence of a peak do not imply the presence of a circle in

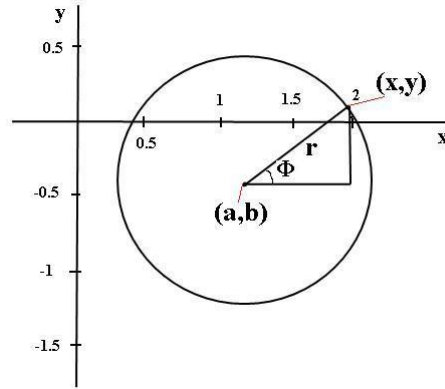


Figure 3.9: Cartesian plane

the image, because the number of pixels may be insufficient to form a circle. For that reason, it is vital to specify for every radius, the number of pixels (value of the peak) requires to get a circle. This is done in accordance with the circumference of a circle given by

$$c = 2 \pi r. \quad (3.3.9)$$

In our work, a circle is selected if at least the 80% of the points composing the circumference are lying on pixels. For our databases, we observe that in any case the 100% of the circle points are lying on the pixels of the digit. According to the size and

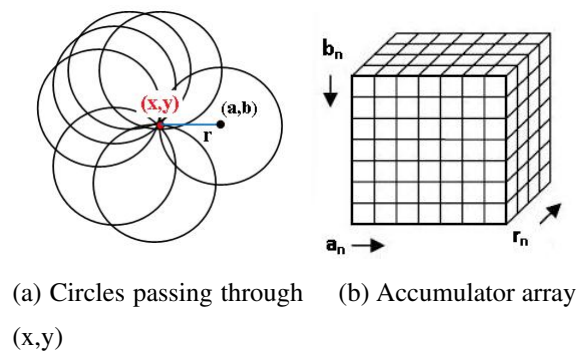


Figure 3.10: Hough transform

shape of the handwritten numbers, we fix the frequent radiuses of circles that can be formed in the digits. We also analyze the alternative locations of those circles in order

to fix the range of values of parameters a and b and select circles with radiuses equal to 4, 5, 6, 7 and 8. Those values are chosen to capture the possible circles contained in digits zero, six, eight and nine. In the case that a circle with $r = 4$ or $r = 5$ is found in the upper part of the digit, another circle with similar radius is searched in the lower part. This searching is done to find the two circles contained in digit eight. Finally, the values of the variables selected for the classification are given by the radius and the coordinates of the start and end-point of the selected circle. That represents a total of 5 variables per image corresponding to the circle detection, 4 values are the coordinates of start and end-points and 1 value is the radius. Examples of circles found in the digits are shown in Figure 3.11. In those cases where the digits do not have circles, these

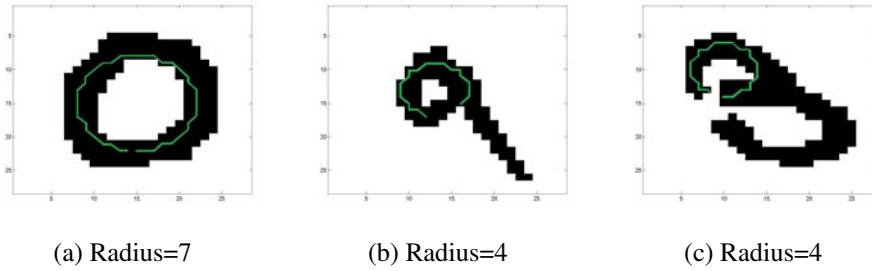


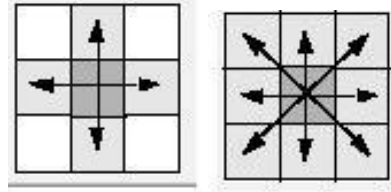
Figure 3.11: Examples of circles

variables assume value zero.

3.3.2. Euler number

The *Euler* number (E) is a measure of the topology of an image, specially used in binary representation. The *Euler* number of binary images can be calculated based on local measures, i.e., from pixel neighborhood relation. Suppose that we consider as neighbors only the four pixels that share an edge (not a corner) with a particular pixel (x, y) . The neighbors are $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, and $(x, y - 1)$. In this situation we have 4-connected neighbors and the connection is defined as 4-connectivity (see Figure 3.12a). An alternative is to consider a pixel as connected not only by pixels on the same row or column, but also by the diagonal pixels. The four 4-connected

pixels plus the diagonal pixels are called 8-connected neighbors, and the connection is defined as 8-connectivity (see Figure 3.12b). The alternative calculation of *Euler*



(a) 4-connectivity (b) 8-connectivity

Figure 3.12: Euler Number

number based on the connectivity is obtained differently in each case (Lin et al. (2006) and Lin et al. (2007)). Denoting as $E(4)$ and $E(8)$ the euler number calculated by 4 or 8-connectivity respectively, then

$$E(4) = \frac{(S_1 - S_3 + 2 \times X)}{4}$$

$$E(8) = \frac{(S_1 - S_3 - 2 \times X)}{4},$$

where S_1 is the number of the following structures in the binary image

0	0	0	0	0	1	1	0
1	0	0	1	0	0	0	0

the S_3 is number of the following structures that are in the binary image

0	1	1	0	1	1	1	1
1	1	1	1	1	0	0	1

and the X is the number of the following structures that are in the binary image

0	1	1	0
1	0	0	1

In our work we use the pixels neighborhood to find the *Euler* Number in 2D images, which is the procedure used in some computational programs. Specifically, we use the 8-connectivity.

3.3.3. Holes

The variable *hole*(H) is specially programmed for this work. It finds holes in the digit and its location up and/or down. If the digit does not have a hole, the variable assumes the value zero. We also consider the case of bigger holes which characterizes the digit zero. In 3.13a the number zero has a bigger hole and in 3.13b the digit has two holes, up and down.

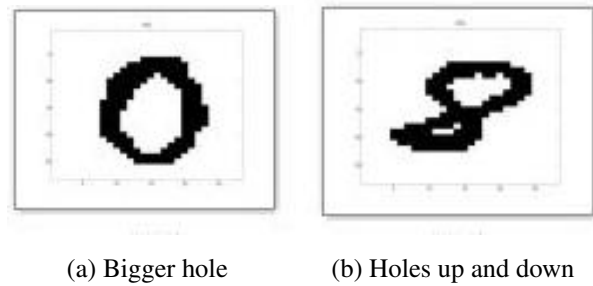


Figure 3.13: Hole variable

3.3.4. Right and left entries

The feature *right entry* (R) is programmed to find if a digit has a right entry and if it is up or down. For example, the number five has a right up entry. The variable *left entry* (L) finds if a digit contains a left entry located up, down or both, like a digit three. If there is no entry, the variable assumes value zero. Examples of these variables are shown in 3.14. In 3.14a the digit has an entry left down and also an entry right up. The digit in 3.14b has two entry by left (up and down) and zero entry in the right. The arrays in the figure indicate the orientation and location of the entries.

3.3.5. Cross in the center

The variable *cross in the center* (C) is defined to detect if a digit has a cross of the digit trace in its center. The shape of a digit with this characteristic is shown in Figure 3.15. The number that contains exactly this shape in the center is digit eight. However,

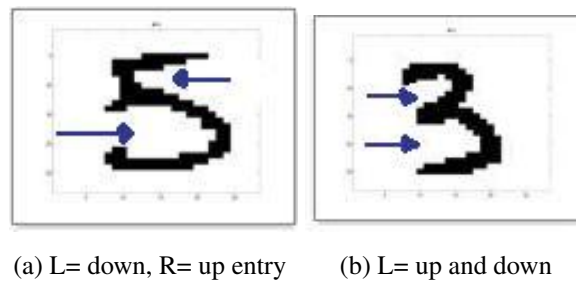


Figure 3.14: Entry variables

we also consider a digit that has a middle cross to upwards or downwards like a nine or six respectively. If the digit does not have cross in the center such as the digit one, the variable assumes value zero.



Figure 3.15: Cross in the center variable

Examples of this feature are shown in Figure 3.16. In (a) the number has a complete cross, in (b) the digit has a middle cross to downwards, and in (c) the nine has a middle cross upwards.

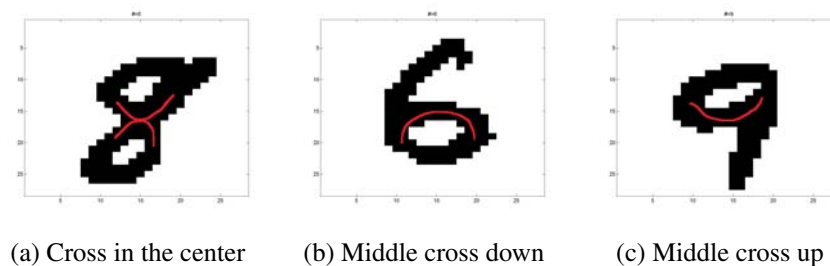


Figure 3.16: Examples of cross in the center variable

3.3.6. Extremes

The variable *extreme* (E) tries to define the contour of a digit by identifying four extreme black pixels of it: the northernmost (E_n), the southernmost (E_s), the easternmost (E_e) and the westernmost (E_w) pixel. If there are more than one pixel occupying one of these extreme location, the pixel situated nearest the central part of the image is chosen. The values of the variable *extreme* are the coordinates of the extreme pixels. As the extreme pixels are four, this variable is characterized by eight values. Examples of this variable are shown in Figure 3.17 where the extreme pixels are depicted in red color.

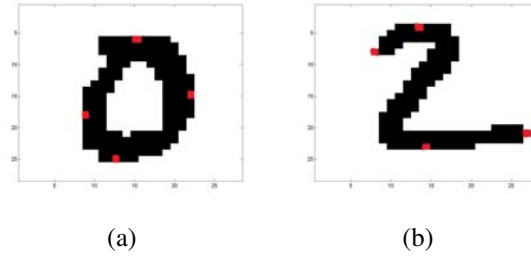


Figure 3.17: Examples of extremes variable

3.3.7. Intersections

The variable *intersections* (I) is obtained considering the extreme pixels E_n, E_s, E_e , and E_w , previously defined. These pixels help to draw two imaginary lines in the image. One of them goes from E_n to E_s and the other line goes from E_e to E_w . Therefore, the *intersections* variable counts the number of times that each imaginary line is intercepted for the trace of the digit (continues trace that define the digit).

Due to we have two imaginary lines, the variable *intersections* has two values, that is the number of intersections of each line. In the example (a) of Figure 3.18 the variable has values (0,0) for both lines, while in example (b) the variable has values (1,1). In Figure 3.18c the values of the variable are (0,1).

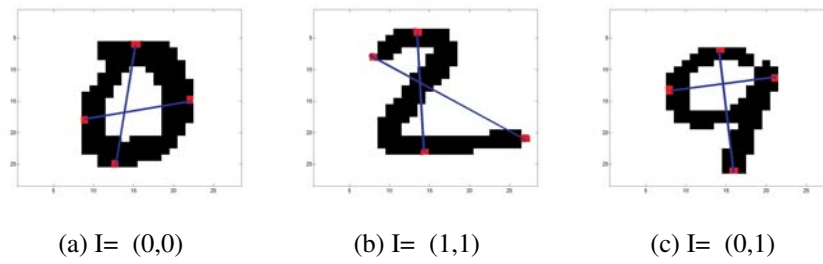


Figure 3.18: Examples of intersection variable

3.3.8. Distance

The *distance* variable is obtained by black pixels located as near as possible to the corners of the image. The corner pixels are the black pixels of the digit located at the northwest corner (co_{NW}), southwest corner (co_{SW}), northeast corner (co_{NE}) and southeast corner (co_{SE}). Two distances are obtained from the coordinates of those pixels, the distance between the northeast and southwest corner pixels, denoted as d_{EW} , and the distance between the northwest and southeast corner pixels denoted as d_{WE} . The distance used is the Euclidean. As results, two values per image are obtained, i.e. $d_{EW} = d(co_{NE}, co_{SW})$ and $d_{WE} = d(co_{NW}, co_{SE})$. Figure 3.19 illustrates the distances of the digit nine.

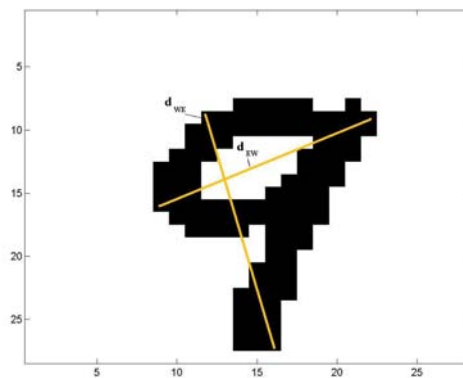


Figure 3.19: Example of distance variable

3.4. Probabilistic classification approach

Once the features that we use in the classification scheme are introduced, we give some details of the two statistical approaches developed in this chapter.

In the first approach the classification is performed by the application of the Bayes' Theorem. Being X the feature vector extracted of an image, the decision rule is stated by deciding that an image belongs to the class c_k if $p(c_k/X) > p(c_j/X)$ for all $k \neq j$. The posterior probability $p(c_k/X)$ is calculated using Bayes' Theorem, as follows

$$p(c_k/X) = \frac{p(X/c_k)p(c_k)}{p(X)}. \quad (3.4.1)$$

Given that the ten classes of digits have almost the same proportion inside the databases, in this dissertation we consider the same prior probability $p(c_k)$. Then, the posterior probability can be expressed as

$$p(c_k/X) \propto p(X/c_k), \quad (3.4.2)$$

where $p(X/c_k)$ is the class-conditioned probability of a feature vector X . This probability is used to find $p(c_k/X)$, i.e., the membership probability to a class c_k of an image in the testing set with feature vector X_{test} . In order to develop an appropriate probabilistic background to find this probabilities, the variables involved in the classification are divided in categorical and quantitative, since we consider that each group requires different technique to be statistically modelled. In the first group we include the variables cross in the center (C), euler (E), hole (H), right entry (R) and left entry (L), forming the feature vector called X_{cat} . In the second group, we consider the variables straight line coordinates (S), diagonal line coordinates (D), straight and diagonal line length (Le), horizontal and vertical intersections (I) and extreme (Ex), composing the feature vector denoted as X_{quant} . The training set is used to find the probability of categorical and quantitative variables of each class expressed in equation 3.4.3. After analyzing the data in training set we can conclude that each group of variables can be treated as independent. In symbols, the probability that an image from training set with feature vector X belongs to a class c_k , is given by

$$P(X/c_k) \propto P(X_{cat}/c_k) \times P(X_{quant}/c_k). \quad (3.4.3)$$

To obtain the class-conditioned probabilities of the categorical feature vector we calculate the joint probabilities of occurrence of the five variables (C, E, H, R, L) for a given class c_k , using a frequentist procedure through the training set. Sometimes, this way of calculation causes a problem due to the lack of observations in the combinations. Besides, the analysis of data suggests that there exists dependence only among some variables. To solve those troubles, we state by cross-validation that the best joint dependence structure for categorical data can be defined by the specific scheme showed in the following equation.

$$P(X_{cat}/c_k) \approx P(L/c_k) \times P(H/L, c_k) \times P(C/L, c_k) \times P(R/C, c_k) \times P(E/R, c_k) \quad (3.4.4)$$

The probability density function of quantitative data (S, D, Le, I, Ex) is modelled by parametric estimation, assuming the multivariate normal distribution to calculate the density inside each class, commonly used for integer-valued features [see [Jain et al. \(2000\)](#)]. The density of an image with a feature quantitative vector X_{quant} given a class c_k , is obtained as follows

$$f(X_{quant}, \mu, \Sigma/c_k) = \frac{1}{\sqrt{|\Sigma_{c_k}|} (2\pi)^d} \exp^{-\frac{1}{2}(x-\mu_{c_k})\Sigma_{c_k}^{-1}(x-\mu_{c_k})'}, \quad (3.4.5)$$

where the mean μ and covariance matrix Σ are estimated from training set in the reference class (equation 3.4.5). The probability density function of each class is valued at the quantitative feature vector for every image in the test set. Moreover, the posterior probability for categorical data in the test set is obtained by equation 3.4.4. Lastly, assuming independence between quantitative and categorical data, the final posterior probability that an image in the test set belongs to a class c_k is given by equation 3.4.3.

The classification process is performed in two stages. In the first stage, we find by cross-validation a cutting point (denoted as p) to classify a subset of the test set. The cutting point value obtained is 0.999. That is, an image with a class probability greater or equal to the cutting point is classified into this class. Otherwise, if none class achieves the 0.999 of probability or more, the image is submitted to a second stage.

In the second stage we have a subgroup of images (called *test2*) that represents the 16,29% of the test set. The procedure in this stage is similar to the previous one.

However, in this case we consider as possible outcomes of each digit in *test2*, only a pair of classes. These classes, denoted as c_i and c_j , are those that have the greatest probabilities of occurrence in the first stage. Thus, we perform the application of Bayesian rule by considering a particular feature vector with the variables considered more discriminant between the classes c_i and c_j in each case. After the process is finished, a digit is classified to a class with greater probability of both. Finally, the test-error rate is calculated achieving a 4,3% for *MNIST* database and a test-error rate of 9,7% in *USPS* dataset. Table 3.4 shows the classification rates for every number.

Table 3.4: Probabilistic approach results

database	0	1	2	3	4	5	6	7	8	9
MNIST	3.57%	2.82%	5.43%	5.05%	6.62%	4.82%	3.86%	7.78%	5.54%	7.53%
USPS	7.0%	8.0%	8.1%	14.5%	18.1%	11.1%	10.3%	9.5%	9.4%	4.0%

According with the results, the procedure has better performance in *MNIST* database than in *USPS*. This behavior could be a consequence of the difficulty of digits in *USPS* to be recognized. The results show that the digits with worse error rate are the seven and nine in *MNIST* dataset and the digit three and four in the *USPS* dataset.

3.5. *K nearest neighbor classification approach*

In this approach, the classification is performed by the *nearest neighbor method* [Cover (1968)], using the variables specially proposed in Section 3.3 for this kind of problems³.

By means of k-nearest-neighbor algorithm, the *training* set feature vectors are used to classify the *test* set. An image is classified by a majority vote of its neighbors, i.e., it is assigned to the class most common among its k nearest neighbors (majority rule). The distance city-block, calculated as the sum of absolute differences, is chosen

³LeCun et al. (1998) and Smith et al. (1994) used all pixels of the image as feature vector, requiring more computer memory and running time

because it provides better performance with integer variables. With respect to the value of the parameter k , there is no consensus in the bibliography in defining the adequate number of k in nearest neighbor classification. Previous works [Hall et al. (2008)] on nearest-neighbor classifiers held the value of k by cross-validation which is the method that we use. Specifically, we choose 5 neighbors. The algorithm is applied to the features defined in Section 3.3 following the sequential forward selection, already explained in Section 2.3, in order to eliminate possible redundant information.

As a results of the sequential selection, 21 variables are included in the classification of *USPS* database and 27 (5 additional different) variables in the *MNIST* database. The excluded features are refereed to redundance information about coordinates of straight and diagonal lines, coordinates of the extreme variable and the information about circles in the image. The five additional variables included in the classification of *MNIST* dataset are some coordinates of the extreme variable. One possible reason to this discrepancy in the number of variables selected could be the fact that the *USPS* set has more heterogeneity in digit shape. Thus, the extremes used in the variable do not work as well as in *MNIST* dataset. The total error rate obtained for the *MNIST* database is 3,65% and 4,39% for the *USPS* dataset. The outcomes for each database are shown in Table 3.5.

Table 3.5: K- nearest neighbors results

database	0	1	2	3	4	5	6	7	8	9
MNIST	1.33%	0.79%	3.20%	5.64%	4.07%	4.15%	1.77%	3.40%	7.08%	5.25%
USPS	0.75%	1.52%	5.9%	6.3%	6.54%	3.91%	4.63%	1.36%	7.66%	4.2%

The results show that the digits with worse error rate in both databases are the three and the eight. It is interesting to observe that in k-nearest neighbor the digits seven and nine have much better error rates than digit eight, while in the probabilistic approach occurs the opposite. This behavior may show a future line of investigation combining both approaches in order to improve results.

3.6. Conclusions and contributions

In general, previous work in handwritten digit recognition contemplate neural network classifiers to perform the classification. In this paper we propose a method based in a multivariate statistical approach. Our study is less sophisticated and obtain competitive results in this area. Other authors work on this database using a baseline nearest neighbor algorithm [see [LeCun et al. \(1998\)](#) and [Smith et al. \(1994\)](#)] to classify the digits directly by the pixels value (784 values per image). We perform the k nearest neighbors technique on the feature vectors of the images (27 values per image) which provides good results on the same dataset and requires less computer memory and recognition time. The proposed variables were specially programmed for this work and they can be easily generalized to be use in any digit database. As the variables are calculated in binary image, we propose a novel method to binarize an image through an optimization procedure that finds the best trace. In addition, we propose an alternative probabilistic approach with similar results than the k nearest rule. Differ from other methods [see [Bottou et al. \(1994\)](#)], the proposed techniques permit to quantify the individual contribution of the variables. They can also be applied easily to different datasets and no changes in the values of the variables are observed by resizing the image.

3.7. Future works

There are some natural extensions to this work that would help expand and strengthen the results.

Firstly, we aim to combine the binarization method proposed in this dissertation with other methods. We observe that our procedure is based on the trace of the digit, while, for instance, the Otsu' method [see [Otsu \(1979\)](#)] is based on minimizing the variances between the blacks and white pixels. We are interested in setting out the binarization problem in terms of a family of thresholds.

Secondly, we aim to analyzed the inclusion of different coefficients of weight for

each variable, in accordance with their contribution in the classification process.

Finally, we are concerned about studying the combination of the classification procedures proposed in section 3.4 and 3.5. We think that the performance of K-nearest neighbor technique could be improved, may be by adding some information about the nearest neighbors [see [Domeniconi et al. \(2005\)](#), [Han et al. \(2001\)](#), [Hastie and Tibshirani \(1996\)](#), [Dudani \(1976\)](#) and [Bailey and Jain \(1978\)](#)].

One extension that would be interesting is to estimate the posterior probability by the application of the K-nearest neighbor classifier [[Atiya \(2005\)](#) and [Fukunaga and Hostetler \(1975\)](#)]. In this regard, we already initiated some studies during the development of this dissertation but without concluding results.

CHAPTER 4

Functional Data Analysis for images

The main goal of this chapter is to build a link between functional data analysis and digital images, in order to explore how this characterization can be useful in classifying or recognizing images. Functional Data Analysis (FDA) has received increased attention during the last decade. It is a vast topic where there are still many open questions. Recent and notable introductions to this application and computational methodologies can be found in [Ramsay and Silverman \(2006\)](#) and [Ferraty and Vieu \(2010\)](#).

Functional data sprung in the last two decades since modern data loggers have permitted to sample and store physical quantities at high frequencies and the Internet made it easy and fast to share high quantity of information. For example, in meteorology: temperatures, pressures and humidities if sampled every minute or every second can be considered functional data. In finance: inter-diary stock prices can be seen as functional. In engineering: per minute electrical energy demand and production is functional. In medicine: measuring a patient temperature, pressure or heart beats count every minute will generate functional data. In environmental sciences we can find functional data sampling a river flow or the amount of CO_2 in the air.

Clearly, data in many fields come to us through a process naturally described as

functional. For instance, a random variable can be observed at different point in time in a range (t_{min}, t_{max}) , where the grid becomes finer and finer meaning that consecutive instant are closer and closer. One way to analyze these data is to consider them as a single observation over a continuum. The continuum is often time, but may also be the spatial location, probability, wavelength, etc.

Similarly, an image can be analyzed as a set of functions where each row (or column) denoted as y_i is a particular function that repeats n times with similar characteristics. Usually, a functional datum for the replication i is seen as a set of discrete measured values y_{i1}, \dots, y_{in} . These values are converted to a function x_i with values $x_i(t)$ calculated for any desired argument value t .

Unfortunately, we rarely receive data in their functions form but they come in tabular format. It is our task to transform all of them into functions, i.e., to transform observations into functions. We propose the use of functional data analysis to reduce image dimension through the extraction of functional principal components (*FPCA*). As a first approximation to the classification problem under the functional data viewpoint, we suggest selecting a group of representative *FPCA* of each image. Then, we use their information as a measure of distance among the groups and perform the classification. The problem of classifying landscape and non-landscape described in Chapter 2 is used to give an example on how the proposed procedure works.

This chapter is organized as follows. In Section 4.1 we describe the basis functions used in the application, to transform observations into functions. We describe the Fourier basis expansion. Then, we analyze a particular case of Wavelet basis that is, the Haar basis. We also give some alternatives to find the best number of K basis functions, according with the smoothing implemented. Section 4.2 is devoted to explain the smoothing and penalization in FDA. In Section 4.3 we describe the functional principal components characteristics. We give in Section 4.3.1 an application to image classification using images scenes from the databases already described in Section 2.1 of Chapter 2. Finally, Section 4.4 provides some concluding remarks and discuss different avenues for future research.

4.1. Basis Functions

Generally, we are concerned about a sample of functional data, rather than just in a single function x . Particularly, the observation of the function x_i might consist on n_i pairs (t_{ij}, y_{ij}) , for $j = 1, \dots, n_i$. Often, the argument values t_{ij} are the same for each record (as it is our case), but this is not always the case since in other contexts those arguments may vary from record to record.

Functions will never be expressed in their closed analytical form, but as a linear combination of a set of predefined functions called "*basis functions*".

Functional data analysis needs data to be represented as a function. The most commonly used strategy is to work with a set of functional building blocks ϕ_k , for $k = 1, \dots, K$, called basis functions. The basis functions are mathematically independent of each other and they have the property to be approximated arbitrarily well to any function, by taking a weighted sum or linear combination of a large number K of these functions. Given a basis functions $\{\phi_1, \phi_2, \dots\}$, we want to create another function $x(t)$ that is the finite linear combinations of all ϕ_i and that approximates well our numerical data. More precisely, given a set of numerical observations y_i that we suppose are taken at time t_i , we want to choose a basis functions $\{\phi_i\}$ $i \in I$, a natural number K and a set of coefficients $\{c_i\}$, $i = 1, \dots, K$, such that

$$\begin{cases} x(t) &= \sum_{k=1}^K c_k \phi_k(t) \\ x(t_i) &\approx y_i. \end{cases} \quad (4.1.1)$$

In matrix notation, we can write

$$x = \mathbf{c}'\Phi = \Phi'\mathbf{c}. \quad (4.1.2)$$

The construction of an actual function becomes a matter of selecting the basis functions and the value of K . The coefficients c_k are obtained in accordance with the selected basis functions, as it is explained in section 4.2.

There are different basis functions commonly used in the literature. Examples of these are: spline, polynomial, Fourier and Wavelet basis functions. One possible criterion to select basis functions is according to the data structure if the data is periodic

or non-periodic. Most applications involving non-periodic data frequently use spline basis functions, while applications which involve periodic data often use Fourier basis. Due to the observed periodicity in the data of image structure, we use in this dissertation Fourier and wavelet basis, as a first analysis of them in functional data context.

4.1.1. Fourier Basis

Fourier basis allows us to decompose any periodic function into the sum of a set of simple oscillating functions with different frequency and width, namely sines and cosines. Following [Ramsay and Silverman \(2006\)](#) notation, the Fourier basis are given by

$$x(t) \approx c_0 + c_1 \sin(r\omega t) + c_2 \cos(r\omega t) + c_3 \sin(r\omega t) + c_4 \cos(r\omega t) + \dots \quad (4.1.3)$$

defined by the basis

$$\begin{cases} \phi_0(t) & 1 \\ \phi_{2r-1}(t) & \sin r\omega t \\ \phi_{2r}(t) & = \cos r\omega t. \end{cases} \quad (4.1.4)$$

where $r = 1, 2, \dots, \frac{\text{period}}{2}$ is the observed number of complete cycles, c_0, \dots, c_k are the elements c_k of the coefficient matrix, i.e., the coefficients of the functional basis. The period is determined by the parameter ω as $\frac{2\pi}{\omega}$. In [Figure 4.1](#) it is shown the decomposition of the Fourier basis with different periods.

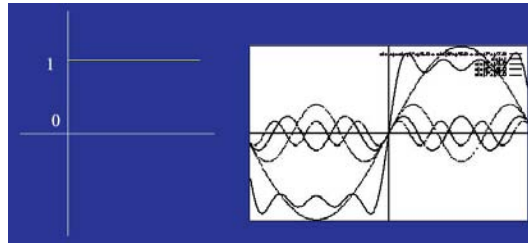


Figure 4.1: Fourier basis

If the values of t are equally spaced on m and the period is equal to the length

of function m , then the basis functions are orthogonal, i.e., the product matrix $\phi' \phi$ is diagonal.

4.1.2. Haar Basis

The Haar basis is the simple case of wavelet basis functions. The term wavelet is referred to an oscillation with an amplitude that starts out at zero, increases, and then decreases back to zero. Wavelets are a family of orthonormal basis functions [see Mallat (2008)] obtained by translation and dilatation of a mother wavelet ψ with $\int \psi(t)dt = 0$ and a father wavelet or scaling function¹ ϕ with $\int \phi(t)dt = 1$. From ψ and ϕ , we can define

$$\psi_{p,q}(t) = \frac{1}{\sqrt{2^p}} \psi\left(\frac{t - 2^p q}{2^p}\right) = \psi_{p,q}(t) = 2^{\frac{p}{2}} \psi(2^p t - q). \quad (4.1.5)$$

and

$$\phi_{p,q}(t) = \frac{1}{\sqrt{2^p}} \phi\left(\frac{t - 2^p q}{2^p}\right) = \phi_{p,q}(t) = 2^{\frac{p}{2}} \phi(2^p t - q). \quad (4.1.6)$$

If we consider a binary scaling of the form $a = 2^p$ and bivalent² translations $b = q 2^p$, where $p = 0, \dots, (n-1)$, and $q = 0, \dots, (2^p - 1)$, we obtain a bivalent orthogonal wavelet basis of the form

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.1.7)$$

and scaling function equals to

$$\phi(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.1.8)$$

Therefore, the Haar wavelet basis expansion (see Figure 4.2) for the function $x(t)$ is given by

$$x(t) \approx c_{00} \phi(t) + \sum_{p=0}^{n-1} \sum_{q=0}^{2^p-1} d_{p,q} \psi_{p,q}(t) \quad (4.1.9)$$

¹Also known as the wide wavelet

²A bivalent translation is given by $q2^p$, that is a multiple integer q of the binary factor 2^p .

where $\phi(t) = 1$ is the scaling function, c_{00} is the scaling coefficient, $d_{p,q}$ are the wavelet coefficients and $\psi_{p,q}$ are the wavelet basis functions. The length of the data function is given by $N = 2^n$, where n can be obtained as $n = \frac{\log N}{\log 2}$ or $n = \log_2 N$. Therefore, a disadvantage of using this kind of basis to image analysis is that it requires images which one of the dimensions is equal to 2^n .

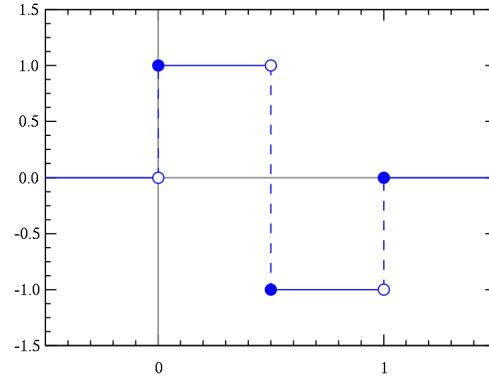


Figure 4.2: Haar wavelet

4.1.3. Number of K basis

There is no consensus about what the number K of basis functions is more appropriate. A number of K too small implies that we may miss some important aspects of the function. Larger K is better, although there is the risk to add noise or variation that we want to avoid. One way to analyze the problem of selecting the number of K is by the minimization of the mean squares error given by

$$MSE[\hat{x}(t)] = Bias^2[\hat{x}(t)] + Var[\hat{x}(t)], \quad (4.1.10)$$

where the bias in estimating $x(t)$ is given by

$$Bias[\hat{x}(t)] = x(t) - E[\hat{x}(t)],$$

and the variance of estimate $x(t)$ is equal to

$$Var[\hat{x}(t)] = E[\{x(t) - E[\hat{x}(t)]\}^2].$$

However, other methods are preferable when the presence of outliers, in particular, the minimization of $E[|x(\hat{t}) - x(t)|]$. Moreover, in literature the stepwise variable selection (see 2.2.4 in chapter 2) is also used to find the adequate number of K , by adding basis functions one after another, testing at each time whether the added function significantly improves fit.

In the cases treated in our thesis, we have images of size $n \times m$, then the maximum number of basis functions is given by the $\min(n, m)$. Moreover, the number of functions x_i is the $\max(n, m)$. A perfect representation of functions x_i is given when K is this maximum.

4.2. Smoothing and penalization

Functional data analysis assumes that the curve being estimated is smoothed, but this is not always true. Two possible strategies are described in the literature in this regards. The first one considers smoothing the raw data and then performing the analysis, whereas the second one decides to leave the noise in the estimated function and then smooth the results of the analysis. We choose to follow the latter method and thus we smooth the results after the applications conducted in this chapter.

The goal in FDA is to fit the discrete observations y_j , for $j = 1, \dots, n$ using the model

$$y_j = x(t_j) + \varepsilon_j \quad (4.2.1)$$

where ε_j is the noise, disturbance or error that contributes a roughness to the raw data. Therefore, the basis function expansion is used to obtain $x(t)$ in matrix notation as

$$x = \mathbf{c}'\Phi = \Phi'\mathbf{c} \quad (4.2.2)$$

where \mathbf{c} is the vector of coefficients c_k with size K , and Φ is the functional vector whose elements are the basis functions ϕ_k . May be one of the tasks in representing raw data as functions is to attempt to filter out the noise as efficiently as possible.

There are three different methods considered [see Ramsay and Silverman (2006)] to perform smoothing. The first one considers to smooth the function fitting the data

by minimizing the sum of squared errors. The second one, to smoothes the function by applying a penalization term, and the third method consists of including a smoothing parameter in the penalization term that regulates the importance of the roughness penalty term. We will briefly explain this techniques in the following subsections.

4.2.1. Smoothing by least squares

A simple linear smoother is obtained if we determine the coefficients of the expansion c_k by minimizing the least squares criterion given by

$$SMSSE(y/c) = \sum_{j=1}^n [y_j - \sum_{k=1}^K c_k \Phi_k(t_j)]^2. \quad (4.2.3)$$

or in matrix term as

$$SMSSE(y/c) = (\mathbf{y} - \Phi \mathbf{c})'(\mathbf{y} - \Phi \mathbf{c}) = \|\mathbf{y} - \Phi \mathbf{c}\|. \quad (4.2.4)$$

Taking derivatives of SMSSE with respect to \mathbf{c} and solving for \mathbf{c} , we can find the estimate coefficient vector $\hat{\mathbf{c}}$ that minimizes SMSSE as follows,

$$\hat{\mathbf{c}} = (\Phi' \Phi)^{-1} \Phi' \mathbf{y}. \quad (4.2.5)$$

The least squares criterion in 4.2.3 is adequate if we assume that the residuals ε_j about the true curve are independently and identically distributed with zero mean and constant variance σ^2 . To deal with nonstationary and/or autocorrelated errors, we may need to bring in a differential weighting of residuals by extending the least squares criterion to the form

$$SMSSE(y/c) = (\mathbf{y} - \Phi \mathbf{c})' \mathbf{W} (\mathbf{y} - \Phi \mathbf{c}). \quad (4.2.6)$$

where \mathbf{W} is a symmetric positive definite matrix. If the variance-covariance matrix Σ_e is known for the residuals ε_j , then

$$\mathbf{W} = \Sigma_e^{-1},$$

Then, the vector of expansion coefficient is equal to

$$\hat{\mathbf{c}} = (\Phi' \mathbf{W} \Phi)^{-1} \Phi' \mathbf{W} \mathbf{y}, \quad (4.2.7)$$

where, as it said, Φ is an n by K matrix that contains the values of the K basis functions at the n sampling points, and y is the vector of discrete data to be smoothed.

4.2.2. Smoothing with penalization

A strong option for approximating discrete data to a function is to use the roughness penalty or regularization approach. A commonly way to quantify the notion of *roughness* is by a penalization term that measures the function's roughness as the integrated squared of the q derivative, i.e.,

$$PEN_q(x) = \int [D^q x(t)]^2 dt, \quad \text{where } q \geq 2.$$

With the introduction of the penalization term, we need to modify the least squares fitting criterion of equation 4.2.6. Let $x(t)$ be the vector resulting from function evaluated at the vector t of argument values. Thus, the least square criterion is defined as

$$SMSSE(y|x) = [y - x(t)]' \mathbf{W} [y - x(t)].$$

The penalized residual sum of squares is defined as

$$PENSSE(y|x) = SMSSE(y|x) + PEN_q(x). \quad (4.2.8)$$

Redefining the roughness penalty $PEN_q(X)$ in matrix term we have

$$PEN(x) = \int [D^q x(t)]^2 dt = c' R c, \quad (4.2.9)$$

where $R = \int D^q \phi(s) D^q \phi'(s) ds$ contains the penalized basis functions and c is the coefficient vector. The coefficient matrix for smoothing with penalization is obtained as

$$\hat{c} = (\Phi' \mathbf{W} \Phi + R)^{-1} \Phi' \mathbf{W} y \quad (4.2.10)$$

4.2.3. Smoothing with parametric penalization

This approach introduces a measure of the function's roughness by including a smoothing parameter in the SMSSE, as follows,

$$PENSSE_{\lambda}(y|x) = SMSSE(y|x) + \lambda PEN(x) \quad (4.2.11)$$

where λ is a smoothing parameter that measures the rate of exchange between fit to the data and variability of the function $x(t)$, quantified by the penalization term $PEN(x)$. The coefficient matrix for smoothing with parametric penalization is obtained as

$$\hat{c} = (\Phi' \mathbf{W} \Phi + \lambda R)^{-1} \Phi' \mathbf{W} y \quad (4.2.12)$$

4.3. Functional Principal Components

Functional Principal Components is a branch of Functional Data Analysis. It is the analog of Multivariate Principal Components but applied to functional data. Put simply, it is a technique that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called *principal components*. The counterpart of variables values x_{ij} in the multivariate context are function values $x(t)$ in the functional context.

In the latter context, the principal component analysis can be defined as the task of finding a linear combination of the functions values, such as

$$f_i = \int \xi(t) x_i(t) dt \quad (4.3.1)$$

where $\xi(t)$ is a weight functions and $x_i(t)$ are the function values. The f_i represents the principal component scores corresponding to the weight function $\xi(t)$.

In functional data analysis, as we explained before, we can apply smoothing methods in preprocessing the data to obtain functional observations and/or we can incorporate it into the results, that is, into the principal component analysis. In *FPCA* the smoothing is somewhat different in the sense that we may include into the function the penalization or regularization term of the estimated principal component curves. That is, we can apply smoothing or penalization to the functional principal components (*FPCA*) by considering the options described in the previous Subsection 4.2.2 and 4.2.2.

In the case of unsmoothed functional PCA, in order to find the components, we used the sample variance PCASV of the principal component scores $\int \xi x_i$ over the observations x_i . Then, the first principal component weight function of equation 4.3.1 is chosen to maximize the PCA sample variance given by

$$PCASV = \text{var}\left(\int \xi_1 x_i(t)\right) = N^{-1} \sum_i \left(\int \xi_1 x_i(t)\right)^2, \quad (4.3.2)$$

subject to:

$$\blacksquare \int \xi_1(t)^2 ds = \int \xi_1^2 = 1.$$

In the case of smoothed functional PCA, we aim to prevent the roughness of the estimated principal component ξ from being too large. Then, the penalized sample variance (PCAPSV) is obtained as

$$PCAPSV(\xi) = \frac{\text{var}(\int \xi x_i)}{\|\xi\|^2 + PEN_p(\xi)}. \quad (4.3.3)$$

Finally, a smoothing parameter λ can be introduced in the penalization term, which regulates the importance of the roughness penalty term. Hence, the first principal component weight function is chosen to maximize the penalized parametric sample variance (PCAPPSV) that is obtained as

$$PCAPPSV(\xi) = \frac{\text{var}(\int \xi x_i)}{\|\xi\|^2 + \lambda PEN_p(\xi)}. \quad (4.3.4)$$

In all applications shown in this chapter we do not smooth the raw data, but smooth the functional principal components.

4.3.1. Application with Fourier basis

As it was explained we perform functional principal components analysis on images in order to reduce their dimensions. We expect that the principal components give us insights into patterns of variation amongst images to perform classification between landscape and non-landscape image scenes (as described in Chapter 2). As we treat the image in gray-level (see Introduction), the values of the functions are gray-color

intensities. Then, the term image variability is referred in this work as the changes of gray intensities.

The first application shown in this section employs the Fourier basis to perform the extraction of the functional principal components. Then, we calculate the *FPCA* for each gray-level image.

Given an image of size $r \times s$, we convert the raw data as functional (see equation 4.1.3), by considering the maximum number of basis functions, that is, $k = \min(r, s)$. Consequently, the number of functions x_i is given by $\max(r, s)$. Next, we extract the number of *FPCA* that jointly explain at least the 90% of the variability in the image. We use different images from the *WLW* and *GLP* databases, already described in Section 2.1, Chapter 2. In the first analysis we do not smooth neither the functional data, nor the principal components. The variability in all images considered, is explained in 90% by four or five components.

Figure 4.3a shows an example of one of the images used to perform the analysis using Fourier basis. The functions in this image are represented by their columns because the $\max(r, s) = s$.

An interesting analysis is to compare the functional principal component obtained without penalization and with the maximum number of K fourier basis functions with the principal components obtained in the multivariate case by considering the Fourier coefficients as the variables. The results of the four principal components extracted to the *Example 1* in both cases are shown in Figures 4.4b and 4.4a.

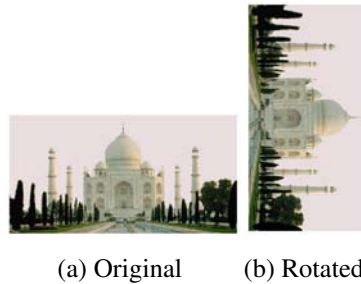
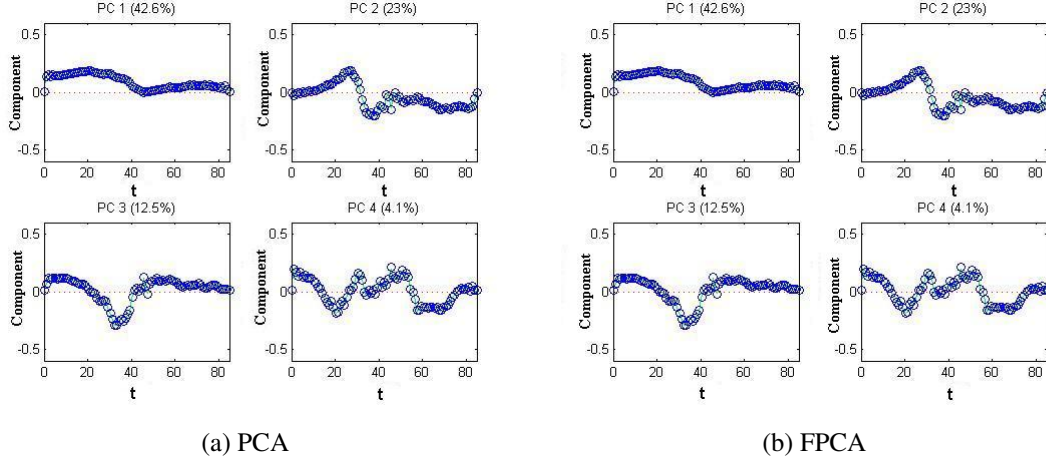


Figure 4.3: Example 1

Figure 4.4: Example I: four *FPCA*-Fourier basis

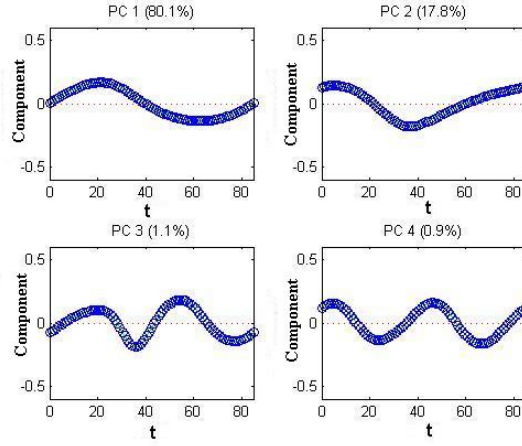
As it can be seen, both plots are identical. This is as a consequence of the orthogonality of the Fourier basis functions, where the coefficient matrix of equation 4.2.5 is reduced to

$$\hat{c} = \Phi' \mathbf{y}. \quad (4.3.5)$$

Then, the *FPCA* without penalization using Fourier basis and the maximum k leads to the same outcomes obtained by applying multivariate *PCA* over the Fourier coefficient matrix.

Next, we include the penalization term in the *FPCA* by penalizing the curves with the second derivative, i.e., with $q = 2$ (see equation 4.2.9). We also include the penalized parameter $\lambda = 0.1$, chosen by cross-validation. The four penalized *FPCA* of the *Example I* are in Figure 4.5. In order to have a better interpretation of the functional principal component curves, we rotate the image as in Figure 4.3b. From the analysis of the *FPCA* of Figure 4.5, we observe that the first component explains the 80% of the total variability.

The first component contains the information about changes of color intensities in the image. Its shape shows that the principal component scores has opposite sign in both sides of the curve. These changes can be observed in the image, since the right side of the *Example I* in Figure 4.3b has brighter intensities than the left side. The

Figure 4.5: Example I: *FPCA* with penalization-Fourier basis

second *FPCA* in Figure 4.5 explains the 17.8% of the variability in the image. The shape of this component seems to show the periodicity present in the image. It may suggest that the difference between the left and right side of the picture, is located in the center of it.

Another example (*Example 2*) is shown in Figure 4.6. For this image we extract four *FPCA* that better explain the behavior of functions x_i in the image. The functions are represented by the rows of the image because the $\max(r, s) = r$. From the analysis



Figure 4.6: Example 2

emerges that the first component explains almost 80% of the variability (see Figure 4.7). The shape of this curve represents the changes of color intensities contained in the image. In this case, the first *FPCA* shows that in the center of the image the color intensities change with respect to the rest of the curve. This variability can be observed

in the image of *Example 2*, where the brighter color of the sun is located at the middle of the landscape. The examples above seem to indicate a possible discriminant power

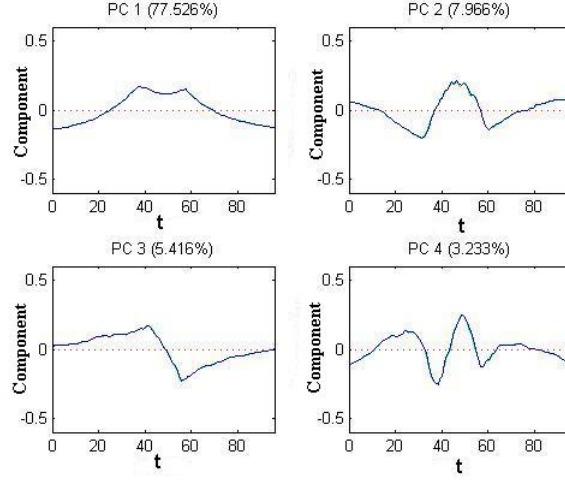


Figure 4.7: Example II: *FPCA* with penalization-Fourier basis

that the functional principal component may contain. Although we consider that further analysis are needed to have a clear conclusion about this issue, specially in terms of penalized functional principal component, in subsection 4.3.3 we conduct a preliminary study about the performance of function principal component as a discriminative variable of classification.

4.3.2. Application with Haar basis

The second application conducted in this chapter involves the Haar basis functions. Due to the structure of these basis, one of the dimension of the image have to be $N = 2^n$ (see Subsection 4.1.2). For that reason, we choose images with size $r \times s = 2^{10} \times 128$. We transform the raw data into functional, applying Haar basis functions, without penalization. We use an arbitrary number of basis functions equals to $K = 0.10 \times \min(r, s)$.

Then, we extract the number of principal component that jointly explains at least the 90% of the total variability in the image. Although this is a simple case, we aim

at comparing results obtained with Fourier basis and with Haar basis functions. The *FPCA* extracted to the image of *Example 3* are shown in Figure 4.9.



Figure 4.8: Example 3

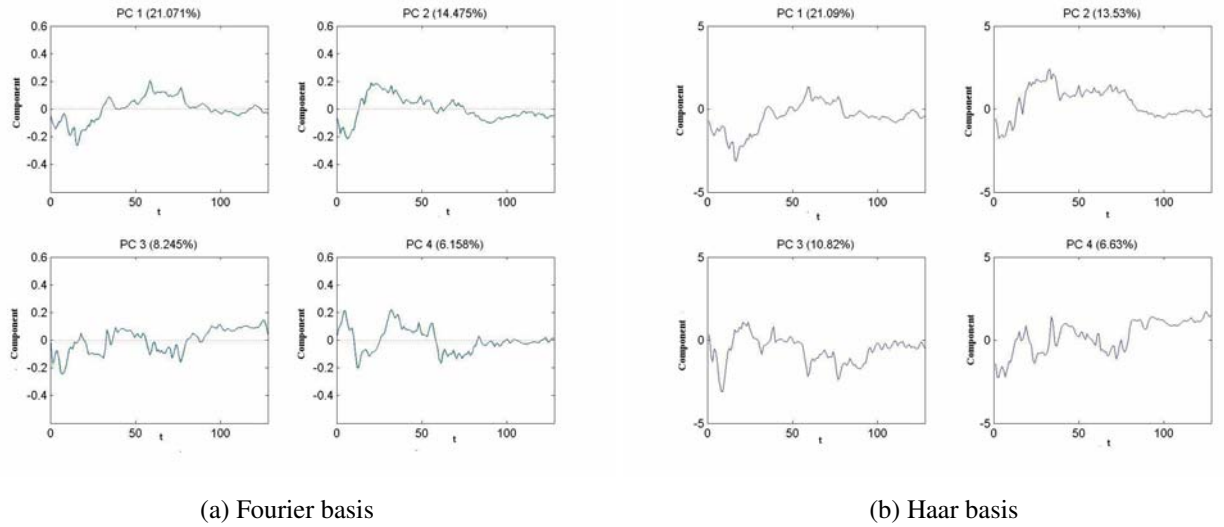


Figure 4.9: Example III: *FPCA*, comparison between Haar and Fourier basis

After a brief preliminary analysis, we observe that there is no evident differences between the *FPCA* extracted through both basis functions. However, we consider that further studies have to be done in this regards, since penalization or complex situations were not taken into account in the analysis.

4.3.3. Classification with functional principal components

In this section we describe a line of investigation, still in progress, to use the FPCA in the classification of digital images. We select a number of components that explain the greatest percentage of variability in the image. Thus, those components are used in the classification process to assign an image to a group.

Images used in this analysis come from the *GLP* database, already described in Section 2.1 Chapter 2. Same size images ($r \times s$) are required in order to have functional components also with same size. Then, 100 images are selected, 52 landscape scenes and 48 non-landscape scenes. In addition, all studies performed in this Section were done with the application of Fourier basis functions.

The procedure that we propose consists of extracting all possible functional principal components of every image to be classified. Then, we select a P number of them as a way to reduce image dimension, in order to perform the classification. In the application described in this chapter, the criterion considered is to choose the *FPCA* that explain the greatest percentage of variability in the image. Specifically, we select $P = 4$ *FPCA* that explain at least the 80% of the variability of each picture.

After selecting the *FPCA*, we work with the principal component to define a distance between the groups. In order to calculate the distances, we first obtain the vector of the median of all *FPCAs* in each group, denoted as f_{pG_i} . We have tried also the mean and trimmean³ to compare results. That is, given two groups, G_1 and G_2 , for every *FPCA* _{p} , $p = 1, \dots, P$, where $P = 4$, we calculate the median function of each component for each group. Figure 4.10 shows the median, mean and trimmean curves obtained for the first *FPCA* extracted from the 100 images. The red line corresponds to landscape pictures (G_1) and blue line corresponds to non-landscape images (G_2). Analyzing the median plot (lower panel of Figure 4.10), we observe that the curves of both groups (landscape or non-landscape) present some significant differences given by the peaks with opposite sign. Moreover, the curves contain noise given by differences between both groups that do not seem to be significant. This behavior is also

³The trimmean or trimmed mean is the mean excluding outliers.

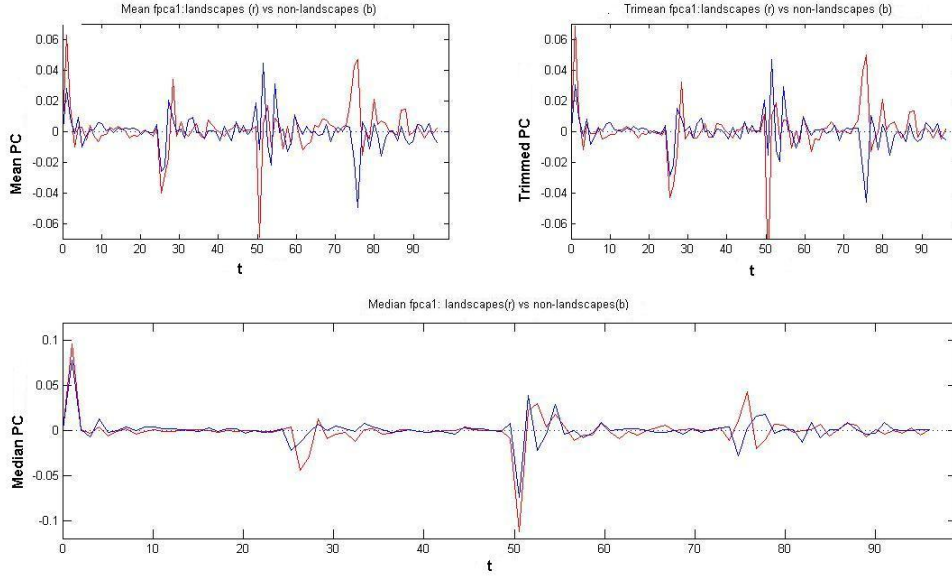


Figure 4.10: First functional principal component: measures

observed in the mean and trimmean plots. Therefore, in order to remove the noise and keep the significant differences between both groups, we redefined the curves in term of their significant values as follow

$$\bar{f}_{pG_i}(h) = \begin{cases} 0 & \text{if } |f_{pG_i}(h) \pm \text{mad}[f_{pG_i}]| < 0 \\ f_{pG_i}(h) & \text{if } |f_{pG_i}(h) \pm \text{mad}[f_{pG_i}]| > 0 \end{cases} \quad (4.3.6)$$

where mad is the median absolute deviation of all values considered to evaluate f_{pG_i} . The resulted curve is called the modified median, denoted as \bar{f}_{pG_i} . The new plots for the first functional principal component are shown in Figure 4.11.

In order to assign each image to the landscape or non-landscape group, we calculate the distance of an image i to each group. For this calculation we introduce a weight ω_p to each $FPCA$, that is the average of the percentage of variability explained for the $FPCA_p$ in each group. Hence, the distance between an image i and the first group (G_1) is given by

$$D_{iG_1} = \sum_{p=1}^P \omega_p d(FPCA_{ip}, \bar{f}_{pG_1}), \quad (4.3.7)$$

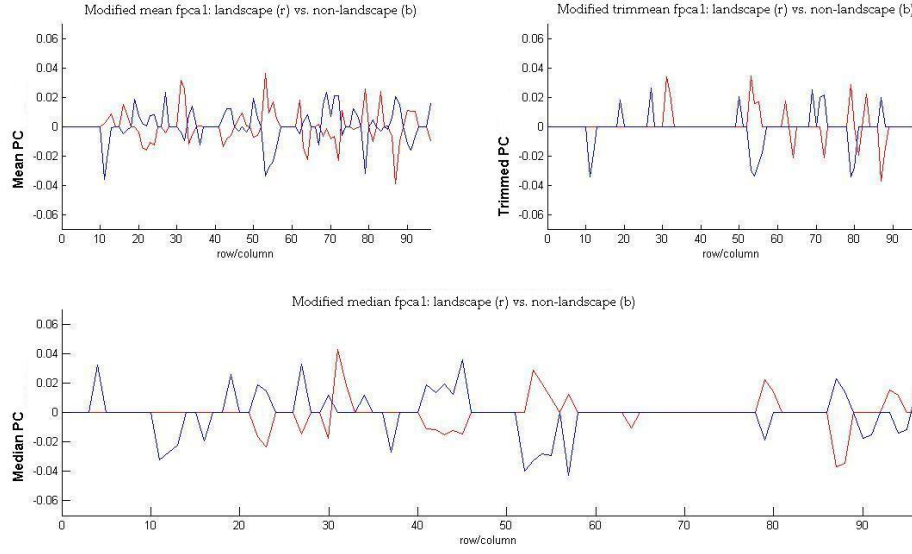


Figure 4.11: First functional principal component: modified measures

and the distance between an image i and the second group (G_2) is

$$D_{iG_2} = \sum_{p=1}^P \omega_p d_i(FPCA_{ip}, \bar{f}_{pG_2}), \quad (4.3.8)$$

where d_i is the Euclidean distance. Finally, an image is classified to group G_1 if

$$D_{iG_1} < D_{iG_2}, \quad (4.3.9)$$

otherwise, it is assigned to group G_2 . The resulted error rate is 29%. Although the error is not good enough, the degree of complexity of this analysis is very low. We are working on improving this outcomes by inspecting some lines of research.

4.4. Conclusions and future works

Functional data analysis is a field relatively unexplored in the past but recently has received increasing attention. Indeed, it represents an area with enormous potential for researchers who want to extend the knowledge on image classification. Our work is

simply a first step in this direction. We propose to continue this study by different lines of investigation.

The use of principal component to perform classification (in the multivariate context) does not necessarily lead to good results in term of classification rate, since the classification success depends on the relative discriminant power of each component to separate as much as possible the groups to be classified. Therefore, an interesting future work could be to choose the functional principal components that maximize the distances among the groups, instead of using those that explains the greater percentage of image variability. With this regards it is also promising to study the functional principal component scores in a multivariate context.

One of the assumption in functional data analysis is that the construction of the functional observations x_i using the discrete data y_{ij} takes place independently for each record i . However, our data, by construction, do not fulfill this assumption. Therefore, other future avenue could explore in depth the classification of images using functional principal component for dependent data [see [Hörmann and Kokoszka \(2010\)](#)].

We think that the use of functional data analysis techniques on other databases with other types of images can shed light on the usefulness of such techniques. In addition, another interesting line for future work (although unrelated to image classification) is the application of these methods in image reconstruction as we believe functional data analysis can be informative in the process of rebuilding an image.

Finally, a more complex future avenue is to study the image as a function $R^2 \rightarrow R$, extending the functional data theory to this context.

CHAPTER 5

Final Conclusions and future works

This thesis deals with different statistical techniques to classify digital images. The large number of applications in this field, ranging from the classical ones such as, medical diagnosis, to the more modern ones such as iris recognition, have attracted considerable research effort with many methods developed in the last few years. In this dissertation, we proposed statistical methods to classify images by their content. In addition, we suggest a group of variables to recognize handwritten digits by their shape. We contribute to the field of image analysis in several ways.

Chapter 2 addresses the classification of images based in their content, an aspect that prior research has been studied without agreement regarding the best method to classify. We propose three features extracted directly from images to discriminate groups. Specifically, we show the application of our proposal to separate landscapes from non-landscapes scenes. Two databases come from different sources are used to conduct the classification. The procedure is carried out by the application of two supervised classifiers, the linear discriminant and the K-nearest neighbors techniques. We achieve an error rate around 3.6%, which is better than the error rate obtained by other authors to similar kind of scene classification. Our method has the benefit to be

intuitive, easy and fast to calculate and uses known techniques. Moreover, we prove that our methodology reports better classification rate than other methods such as, support vector machine technique. We applied both techniques to the same databases and showed that our methods is superior as it classifies with an error rate around 4% while support vector machine offers a 12.4% error-rate. Our results suggest that, K-nearest neighbor technique and linear discriminant classification might be a relevant techniques in the classification of images by their content.

With regards to future works in scene classification, we aim at exploring in depth the analysis of the spectral density to discriminate images with different contents, such as textures. In addition, we are interested in applied our procedure to other databases.

Chapter 3 deals with handwritten digit classification. We suggest the calculation of variables that detect the shape and geometry of numbers. All the variables used in the classification were specially programmed for this work. Since the features are calculated using binary images, we propose a novel method to find an optimum threshold to binarize them. This methodology is concerned about finding the threshold that minimize the variability in the trace of the digit. We have worked with two well-known databases, the *MNIST* and the *USPS* applying different statistical approaches. The first one, the multivariate approach, is based on the application of the K-nearest neighbors algorithm. The second one is a probabilistic approach that involves the use of the Bayes' theorem. We achieve a classification rate around 3.5%. The main contribution of this Chapter is that the procedure we propose is intuitive and easy to be generalized to other digits databases or a images with different sizes with competitive classification rates.

We have several lines of research already in progress to further explore and advance knowledge to handwritten digit recognition. In particular we are interested in including weights coefficients for the variables used in the classification, that will be selected in accordance with their discriminative power. We are also concerned about improving the classification procedure by the combination of the probabilistic and multivariate approaches. One extension in which we are interested is on estimating the posterior probability by the application of the K-nearest neighbor classifier. We believe that this

line of research will produce more robust results.

In Chapter 4 we describe a line of investigation related with the use of functional data analysis to classify images. We first propose to reduce image dimension by the extraction of functional principal components of them. Then, we selected those component that best discriminate the images of the groups, instead of those that best explain the variability of the images. Finally, by using the more useful functional principal component we perform the classification considering a previously defined distance or semi distance. Due to this study is still in progress we do not have concluding results. However preliminary results are promising and we are committed to pursue this line of inquiry.

Our work has also implication for practice. Perhaps one of the most appealing characteristic of this field of study is its practical application in various areas such as medicine (e.g., diagnosis through images), security (e.g., face and iris recognition), finance (e.g. detection of illegal bills) and communication (e.g., handwritten characters recognition). Indeed, recent newspaper articles has emphasize the opportunities and challenges of image classification. For instance, a recent article in a Spanish newspaper (El País - June 14th) informed that the Internet search company Google was launching its application to search images by content including colors and textures (as opposed to its traditional service that searches images by keywords or labels) but it was severely limited because the system is not able to identify objects. Practical applications of our methods can add value to different companies and organizations that deal with image classification dilemmas.

Summarizing, our work has provided more nuances to the understanding of image classification by applying different statistical methods and developing different statistical features. Yet, this a field where there is a lot to be done and learnt. Challenges like improving methodologies or exploring potential classification tools are just a few of the many that remain. It is our future responsibility as scholars to address them successfully.

Bibliography

- Atiya, A. F. (2005). Estimating the posterior probabilities using the k-nearest neighbor rule. *Neural Computation* 17, 731–740.
- Ayers, B. and M. Boutell (2007). Home interior classification using sift keypoint histograms. In *Computer Vision and Pattern Recognition*.
- Bailey, T. and A. K. Jain (1978). A note on distance-weighted k-nearest neighbor rules. *IEEE Transactions on Systems, Man and Cybernetics* 8(4), 311 –313.
- Ballard, D. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition* 13(2), 111 –122.
- Belongie, S., C. Carson, H. Greenspan, and J. Malik (1997). Recognition of images in large databases using a learning framework. Technical report, U.C. Berkeley.
- Benito, M. (2006). *Técnicas Estadísticas para el Análisis de Imágenes*. Ph. D. thesis, Universidad Carlos III de Madrid.
- Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B-48*, 259–302.

- Bosch, A., X. Muñoz, and R. Martí (2007). Which is the best way to organize-classify images by content. *Image and Vision Computing* 25(6), 778 – 791.
- Bosch, A., A. Zisserman, and X. Munoz (2007). Image Classification using Random Forests and Ferns. pp. 1–8.
- Boser, B. E. and et al. (1992). A training algorithm for optimal margin classifiers. pp. 144–152. ACM Press.
- Bottou, E. and V. Vapnik (1992). Local learning algorithms. *Neural Computation* 4, 888–900.
- Bottou, L., C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik (1994). Comparison of classifier method: A case study in handwritten digits recognition. *Pattern Recognition* 2.
- Boutell, M. R., J. Luo, X. Shen, and C. M. Brown (2004). Learning multi-label scene classification. *Pattern Recognition* 37, 1757 – 1771.
- Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations*. Cambridge University Press.
- Castleman, K. R. (1995). *Digital Image Processing* (2 ed.). Prentice-Hall.
- Ciresan, D. C., U. Meier, L. M. Gambardella, and J. Schmidhuber (2010). Deep big simple neural nets excel on handwritten digit recognition. *Neural Computation* 22.
- Cortes, C. and V. Vapnik (1995). Support-vector networks. In *Machine Learning*, pp. 273–297.
- Cover, T. (1968). Estimation by the nearest neighbor rule. *Information Theory, IEEE Transactions on* 14(1), 50 – 55.
- Cross, G. R. and A. K. Jain (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5*, 25–39.

- Decoste, D. and B. Scholkopf (2002). Training invariant support vector machines. *46*, 161–190.
- Domeniconi, C., D. Gunopulos, and P. Jing (2005). Large margin nearest neighbor classifiers. *Neural Networks, IEEE Transactions on* 16(4), 899–909.
- Duda, R. and P. Hart (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM* 15(1), 11–15.
- Duda, R. O. and P. E. Har (1973). *Pattern Classification and Scene Analysis*. Wiley-Interscience.
- Duda, R. O., P. E. Har, and D. G. Stork (2000). *Pattern Classification* (2 ed.). Wiley-Interscience.
- Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics, SMC-6*(4), 325–327.
- Ferraty, F. and P. Vieu (2010). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer.
- Florindo, J. B., M. de Castro, and O. M. Bruno (2010). Enhancing multiscale fractal descriptors using functional data analysis. *I. J. Bifurcation and Chaos*, 3443–3460.
- Fukunaga, K. and L. Hostetler (1975). K-nearest-neighbor bayes-risk estimation. *IEEE Transactions on Information Theory* 21(3), 285–293.
- German, S. and D. Geman (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*, 721–741.
- Gómez, J. T. (2009). *Análisis Comparativo de Algoritmos en Segmentación de Iris*. Ph. D. thesis, Universidad Carlos III de Madrid.
- Gonzalez, R., R. Woods, and S. Eddins (2004). *Digital Image Processing using MATLAB*.

- Hall, P., B. Park, and R. Samworth (2008). Choice of neighbor order in nearest-neighbor classification. *Annals Statistics* 36(5), 2135–2152.
- Han, E.-H., G. Karypis, and V. Kumar (2001). Text categorization using weight adjusted k-nearest neighbor classification. In *Proceedings of Conference on Knowledge Discovery and Data Mining*, pp. 53–65.
- Hastie, T. and R. Tibshirani (1996). Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(6), 607–616.
- Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- Hough, P. V. C. (1962). Method and means for recognizing complex patterns.
- Hörmann, S. and P. Kokoszka (2010). Weakly dependent functional data. *The Annals of Statistics* 38(3), 1845–1884.
- Huang, H.-Y., W.-S. Shih, and W.-H. Hsu (2007). A film classifier based on low-level visual features. pp. 465–468.
- Jain, A. K., R. P. W. Duin, and J. Mao (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1), 4–37.
- John, G. H., R. Kohavi, and K. Pfleger (1994). Irrelevant features and the subset selection problem. pp. 121–129. Morgan Kaufmann.
- Karegowda, A. G., M.A.Jayaram, and A. Manjunath (2010). Feature subset selection problem using wrapper approach in supervised learning. *International Journal of Computer Applications* 1(7), 13–17.
- Keysers, D., T. Deselaers, C. Gollan, and H. Ney (2007). Deformation models for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(8), 1422–1435.

- Kohavi, R. and G. H. John (1997). Wrappers for feature subset selection. *Artificial Intelligence* 97(1-2), 273 – 324.
- Lauer, F., C. Suen, and G. Bloch (2007). A trainable feature extractor for handwritten digit recognition. *Pattern Recognition* 40(6), 1816–1824.
- LeCun, Y., B. Boser, J. S. Denker, R. E. Howard, W. Hubbard, L. D. Jackel, and D. Henderson (1990). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems* 2, 396–404.
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278–2324.
- Lin, X., J. Ji, and Y. Gu (2007). The euler number study of image and its application. pp. 910 –912.
- Lin, X., Y. Sha, J. Ji, and Y. Wang (2006). A proof of image euler number formula. *Science in China Series F: Information Sciences* 49, 364–371.
- Liu, X., L. Zhang, M. Li, H. Zhang, and D. Wang (2004). Boosting image classification with lda-based feature combination for digital photograph management. pp. 887–901.
- Liu, Y., D. Zhang, G. Lu, and W.-Y. Ma (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition* 40(1), 262 – 282.
- Luo, J. and A. Savakis (2001). Indoor vs outdoor classification of consumer photographs using low-level and semantic features. Volume 2, pp. 745 –748.
- Mallat, S. (2008). *A Wavelet Tour of Signal Processing* (2 ed.). Academic Press.
- Nandgaonkar, S., R. Jagtap, P. Anarase, B. Khadake, and A. Betale (2010). Image mining of textual images using low-level image features. Volume 9, pp. 588 –592.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics* 9(1), 62–66.

- Park, S. B., J. W. Lee, and S. K. Kim (2004). Content-based image classification using a neural network. *Pattern Recognition Letters* 25(3), 287 – 300.
- Patino-Escarcina, R. and J. Ferreira Costa (2008). The semantic clustering of images and its relation with low level color features. pp. 74 –79.
- Peña, D. and J. Rodriguez (2003). Descriptive measures of multivariate scatter and linear dependence. *Journal of Multivariate Analysis*. 2(58), 361–374.
- Pratt, W. K. (1991). *Digital Image Processing*. Pearson Prentice Hall.
- Qin, J. and N. H. Yung (2010). Scene categorization via contextual visual words. *Pattern Recognition* 43(5), 1874 – 1888.
- Ramsay, J. and B. Silverman (2006). *Functional Data Analysis*. Springer.
- Ripley, B. D. (2004). *Spatial Statistics*. Wiley-Interscience.
- Serra, J. (1984). *Image analysis and mathematical morphology*, Volume 1. Academic Press.
- Serrano, N., A. Savakis, and J. Luo (2002). A computationally efficient approach to indoor/outdoor scene classification. *International Conference on Pattern Recognition* 4.
- Shah, S. and V. Gandhi (2004). Image classification based on textural features using artificial neural network. *The Institution of Engineers Journal*.
- Shumway, R. and D. S. Stoffer (2010). *Time Series Analysis and its applications. With R examples* (3 ed.). Springer.
- Simard, P., Y. LeCun, and J. Denker (1993). Efficient pattern recognition using a new transformation distance. *Advances in Neural Information Processing Systems* 5, 50–58.

- Smith, S., M. Bourgoïn, K. Sims, and H. Voorhees (1994). Handwritten character classification using nearest neighbor in large databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(9), 915 –919.
- Szumner, M. and R. W. Picard (1998). Indoor-outdoor image classification. pp. 42–51.
- Thulasiraman, P. (2005). *Semantic classification of rural and urban images using learning vector quantization*. Ph. D. thesis, Louisiana State University.
- Vailaya, A., A. Jain, and H. J. Zhang (1998). On image classification: City images vs. landscapes. *Pattern Recognition* 31, 1921–1935.
- Vailaya, A., A. Member, M. A. T. Figueiredo, A. K. Jain, H.-J. Zhang, and S. Member (2001). Image classification for content-based indexing. *IEEE Transactions on Image Processing* 10, 117–130.
- Vapnik, V. and A. Lerner (1963). Pattern Recognition using Generalized Portrait Method. *Automation and Remote Control* 24.
- Varma, M. (2007). Learning the discriminative powerinvariance trade-off.
- Wang, J., J. Li, and G. Wiederhold (2001). Semantics sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(9).
- Yuchun, L. (1991). Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks. *Neural Computation* 3(3), 440–449.
- Zhang, J., S. Lazebnik, and C. Schmid (2007). Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision* 73, 2007.